ISBN 978-99990-0-043-7

Federal Democratic Republic of Ethiopia Ministry of Education

INFORMATION TECHNOLOGY STUDENT TEXTBOOK GRADE 12

FDRE MOE

INFORMATION TECHNOLOGY

STUDENT TEXTBOOK

GRADE 12

Federal Democratic Republic of Ethiopia Ministry of Education

Take Good Care of This Textbook

This textbook is the property of your school.
Take good care not to damage or lose it.
Here are 10 ideas to help take care of the book:

1. Cover the book with protective material, such as plastic, old newspapers or magazines.
2. Always keep the book in a clean dry place.
3. Be sure your hands are clean when you use the book.
4. Do not write on the cover or inside pages.
5. Use a piece of paper or cardboard as a bookmark.
6. Never tear or cut out any pictures or pages.
7. Repair any torn pages with paste or tape.
8. Pack the book carefully when you place it in your school bag.
9. Handle the book with care when passing it to another person.
10. When using a new book for the first time, lay it on its back. Open only a few pages at a time. Press lightly along the bound edge as you turn the pages. This will keep the cover in good condition

INFORMATION TECHNOLOGY

STUDENT TEXTBOOK

GRADE 12

Writers:

Teshome Alemu Wedajo (MSc)

Wendwesen Endale Workneh (MSc)

Editors:

Gezahagn Gebeyehu Gamu (MA) (Curriculum Editor)

Berhane Woldegabriel Zeweldi (MSc) (Content Editor)

Temesgen Debelo Ayana (MA) (Language Editor)

Illustrator:

Selamneh Belete Tekle (MSc)

Designer:

Dagnachew Melesew Alemayehu (MSc)

FEDERAL DEMOCRATIC REPUBLIC OF ETHIOPIA

MINISTRY OF EDUCATION

HAWASSA UNIVERSITY

First Published August 2023 by the Federal Democratic Republic of Ethiopia, Ministry of Education, under the General Education Quality Improvement Program for Equity (GEQIP-E) supported by the World Bank, UK's Department for International Development/DFID-now merged with the Foreign, Common wealth and Development Office/FCDO, Finland Ministry for Foreign Affairs, the Royal Norwegian Embassy, United Nations Children's Fund/ UNICEF), the Global Partnership for Education (GPE), and Danish Ministry of Foreign Affairs, through a Multi Donor Trust Fund.

The Ministry of Education wishes to thank the many individuals, groups and other bodies involved – directly or indirectly – in publishing this Textbook. Special thanks are due to Hawassa University for their huge contribution in the development of this textbook in collaboration with Addis Ababa University, Bahir Dar University and Jimma University.

Printed by:

GRAVITY GROUP IND LLC

13th Industrial Area,

Sharjah, UNITED ARAB EMIRATES

Under Ministry of Education Contract no. MOE/GEQIP-E/LICB/G-01/23

ISBN:978-99990-0-043-7

Table of Contents

Unit Six

Fundamentals of Programming 164

UNIT 1

INFORMATION SYSTEMS AND THEIR APPLICATIONS

Learning Outcomes

At the end of this unit, students will be able to:

- Explain ethical and legal issues in information systems
- Explain environmental, health, and social issues in information system
- Recognize the basics of intellectual property rights
- Describe digital identity management
- Appreciate the role of digital technologies in citizenship

Unit Overview

Information systems have had an impact far beyond the world of business. New technologies create new situations that we have never dealt with before. The introduction of new technology can have a profound effect on human behavior. New technologies also give us capabilities that we did not have before, which in turn, create environments and situations that have not been specifically addressed in ethical terms. Users should also be aware of the consequences and implications of information systems' use and access on the one hand and maintaining the rights of creators and innovators on the other. This unit deals with the fundamentals of ethical, legal, social, environmental and health issues, intellectual property rights, digital identity management, and the role of digital technologies in citizens' engagement.

1.1. Ethical, Legal, Social, Environmental and Health Issues in the Use of Information Systems

The use of new technologies including information systems, has a significant impact on human behavior and our daily activities including how we live, work, and learn. Increased consequences have been emerging which needs all user's awareness about the use of information systems. The knowledge and skills about raising issues and proper use of information systems help students to become responsible members of society who can navigate a knowledge based and technology-led economy. This sub unit will focus on the major issues in use of information systems.

1.1.1 Ethical and Legal Issues in Information System

Brainstorming 1.1

- How could someone gain access to online banking account?
- What would be the consequences of such access?

Ethics is all about the principles of making right or wrong decisions. According to Britannica dictionary, ethics defined as, the discipline concerned with what is morally good and bad and morally right and wrong. Ethical principles needs to be respected and if not might result in legal consequences for those cases who are against the applied laws. The role of ethics in information systems has become more crucial as a result of technology. Questions related to ethics and legality are essential in many industries. Many professionals including medical and education professionals, government officials, and business people have legal and ethical oversight to control how their professions function. Information systems, by contrast, lack standard that align with ethical concerns in the current frequently changing and dynamic technological ecosystem. The ethical and legal issues include privacy, security, information gathering and digital ownership related concerns. You will learn about some of the main ethical and legal issues you need to know about as you participate in a constantly changing the digital world.

The following are some ethical and legal issues confronting the current digital world.

Privacy: In today's digitized environment, information privacy relates to personal data stored on the computer system, has become a prime concern for everybody. Most people have their data spread throughout the digital systems, and private information should be accessed using personal passwords. However, personal accounts such as email, social media, and private bank accounts, can be accessed by unauthorized sources which create both ethical and legal concerns. It is morally wrong and unethical to view an individual's private information in anyways which may also end up being a legal issue. All practices and attempts in organizations to check their employee's computer habits also need to be implemented without incurring any damage to individual privacy.

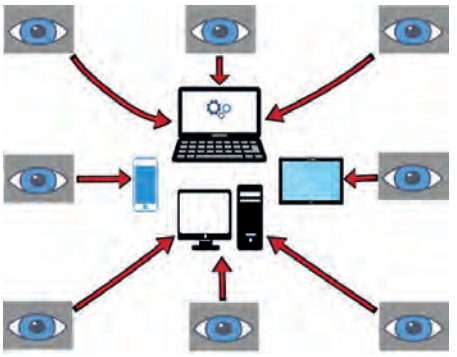


Figure 1.1 Privacy Issues

Digital Ownership: It is ability to access and/or use and protect data, information, and knowledge about ourselves or things we own. The expansion of digital mediums has allowed information to flow more freely than before which makes content easily copied and pasted online. This makes digital ownership hard to control in the digital world. It is unethical to use somebody else's digital works without getting permission from the owner first since it ignores the right of the creator. Creators has right to allow or deny the use of the contents. This permission can promote sharing of knowledge and information, credits the source and protect the consumer from false information. Users resources use should consider such concerns to promote proper use and dissemination of information, and to protect themselves from legal consequences that arises from violating intellectual property ownership rights.

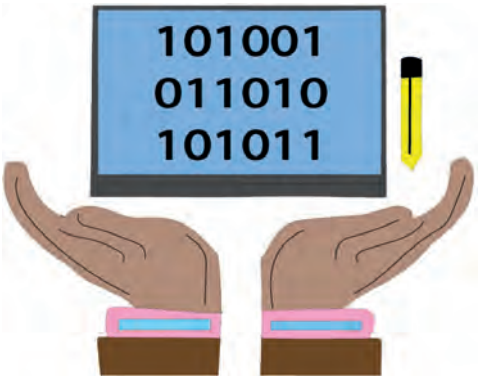


Figure 1.2: Digital Ownership

Links

More information about intellectual property and related concepts in detail in Section 1.2. of this book.

Information Gathering: Many information system applications collect users' data and credentials as part of their operations. Such information gathering trends can lead to many ethical and legal issues depending on the purpose and level of information gathering approaches. The measure has opened a debate about what and why information can be gathered. This issue raises a question of consent which protect individuals against attempts to steal information.

Figure 1.3: Information Gathering

Users need to understand and examine the intention of any information-gathering request by any service providers or third-party applications while they use the services. Such requests need special attention if those organizations, systems, or applications request personal information and credentials. This protects against any illegal and fraudulent information stealing attempts.

Information Security and Liability: Growing amounts of personal and financial information are being transferred and stored online as organizations and businesses depend more and more on electronic data and computer networks to carry out their daily operations. If and when a breach in information security happens, this might expose organizations and businesses to potential security liability. Such attempts might lead to identity theft, which happens when the personal details of innocent people to be harvested by a third party so that they can be used for malicious purposes. The increasing use of online tools and platforms such as blogs and instant messaging is another source of potential liability. Accepted procedures and rules are needed to provide conditions to increase the reliability of information systems. Users also need to report any security breach incidents that happened on their systems and tools to the concerned department/unit or expert in their organizations or the concerned government body promptly.

For example – in Ethiopia, government and private organizations should report any cyber security breaches or attacks to the designated government unit that is responsible for protecting our country's information and information infrastructure from harm.

The above-mentioned pressing ethical and legal issues are assumed to be evolving as technologies change over time, and it is recommended to assess and analyze issues over time, concerning ethical and legal aspects.

Activity 1.1

1. Do you agree that government agencies should monitor information usage and gather citizens' information for national security purposes in the current digital world? Give reasons for your opinion.
2. Find information about the current government agency that is currently responsible for protecting our country's information and information infrastructure from harm? Discuss the procedures that followed for reporting any security incidents to the concerned body.

1.1.2 Social Issues in Information System

Brainstorming 1.2

- What impacts do the internet and information systems have on our social lives?

The expansion of accessible information system technologies has led to several significant changes in how society functions. With personal computers and the Internet offering unrestricted access to all of the knowledge ever created and discovered by humans, the majority of these changes have improvements to society. Those social advantages including access to technology which fills the digital divide gaps within the global citizens, use of advanced technologies like Artificial Intelligence and Augmented Reality in public which changes how peoples act and react , and widespread use of digital platforms , social medias and computer games for business and entertainment activities .

However, other less uplifting social problems have been directly caused by technical advancements of the technologies. Hence, it is crucial to examine and assess the impacts they have to take actions to better understand and mitigate their negative impacts while maximizing the benefits. The following are some negative social issues to consider while using technological services and Internet-based applications.

Socializations Gaps

Figure 1.5 Communication breakdown among families

Socializing within a family unit has always been important, as it strengthens the bonds among family members or friends and ensures cohesion within the group. However, with more and more households owning several computers and numerous portable devices granting access to information and entertainment, some argue that this is leading to a lack of family socialization and natural interactions in the present moment (See Figure 1.5).

It is also become common to see friends who are out in a cafe and still staring at a touchscreen rather than talking to one another. Increase your digital well-being by allowing technology to improve your life and not become a distraction from your life and others.

Cyber Bullying- It is a practice of sending, posting, or disseminating unfavorable, hurtful, or malicious content about someone else online. It can also involve disclosing sensitive or private information about another individual in a way that causes embarrassment or humiliation.

Cyberbullying can happen online through social media, forums, or gaming where users can interact with, or exchange content with others. It can also happen through text and other messaging applications (Figure 1.6).

Figure 1.6 Cyber Bulling

Example - The availability of the internet and computer access can cause problems in terms of online gender-based violence's, harassments and sexual abuse which is one of serious issues to address to protect our community to live a healthy life style.

Social Media and Gaming Addiction - According to addiction center definition social media and gaming addiction is a behavioral addiction that is characterized as being overly concerned about social media and computer games, driven by an uncontrollable urge to log on to or use social media, and devoting so much time and effort to social media that it impairs other important life areas (See Figure 1.7). This day's social media and video gaming are common ways in which people spend their leisure time. Researches on social networking addiction and Internet gaming disorders finds that intensive online activity can negatively impact young people's mental health.

Figure 1.7 Social media Addiction

Students need to be curious and mindful about their use of social media and video games to live a safe and healthy life. Such practices have direct links and impacts on their family relationships, learning performance, future life goals, and mental health.

Plagiarism – Computers and information systems are known for their educational benefits, which are praised, but having access to all knowledge has drawbacks of its own. According to university of oxford, Plagiarism is presenting work or ideas from another source as your own, with or without consent of the original author, by incorporating it into your work without full acknowledgement.

Figure 1.8: Plagiarism

The act of plagiarism has grown to be a significant issue because students can easily copy and paste entire passages of material from Internet resources without giving credit to the author (See Figure 1.8). This has particularly become a challenge for educational institutions, now applying different tools and systems to check the originality of students' work at all levels. Plagiarism in the workplace and in academia can destroy an author's credibility and have significant consequences, such as dismissal from school or job loss. Students need to produce original works, use books and Internet resources as references and sources of insight, and give appropriate credit to authors when they refer to their works in any way.

Cybercrime - Computers have contributed positively to the development of both individual actions and large-scale global movements. The other side of the coin is organized cybercrime which is use of a computer as an instrument to further illegal ends, such as committing fraud, stealing identities, or violating privacy and intellectual property rights as stated in Britannica dictionary. Cybercrime, especially through the Internet, has grown in importance as the computer has become central to education, commerce, entertainment, and government.

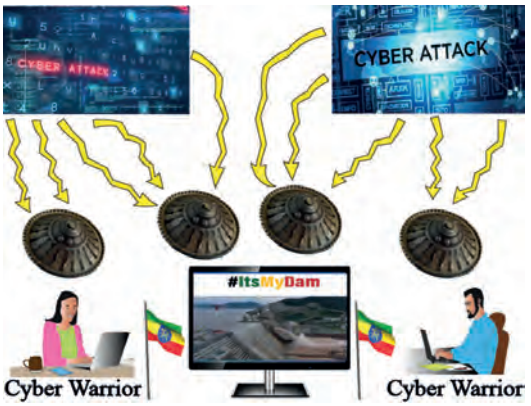


Fig 1.9 Cyber Warrior

According to Official reports from Ethiopian Information Network Security Agency (INSA) officials Ethiopia is also experiencing different cyber-attacks which shows increasing trend from time to time that needs a serious attention and a collaborative effort from all citizens and professionals. This clearly shows that students need to prepare themselves for the 21st-century patriotism mission to protect personal, organizational and national digital resources by being mindful system users to protect against crime and terrorism attempts to ensure the country's digital sovereignty.

Notes

- Interested and skilled students can also start the journey of being skilled cyber warrior who is a well-trained, skilled professional in the information technology or information security industry that manages computer networks and system operations

Job Opportunities and Challenges - The current trends in automation and emerging technologies like artificial intelligent machines, help to automate routine tasks which lead to improved quality of service. On the other hand, such expansion of automated systems leads to loss of jobs because some types of jobs like cashiers, data entry clerks, and factory workers can be managed by automated systems (See Figure 1.10).

Automation often creates as many jobs as it destroys over time. The type of skills that people need to acquire is constantly changing and people with low educational backgrounds are most at risk since the future job market needs more skilled professionals.

We must be mindful of our actions and be responsible for every step that we take to get benefits out of those advancements. We must also be concerned about Internet usage and associated access practices to mitigate social and psychological challenges

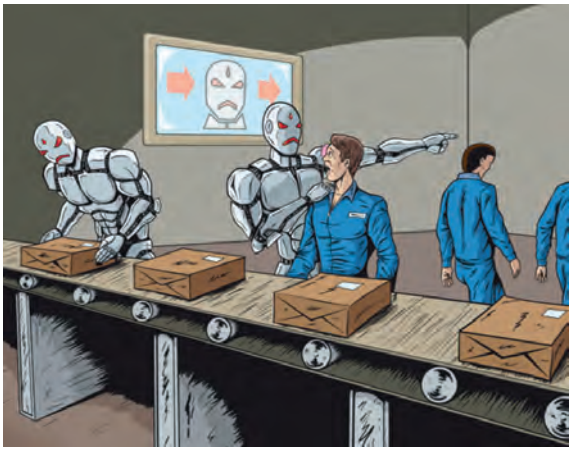


Figure 1.10: Automation would lead to loss of selected jobs

Activity 1.2

1. Discuss the influence of inappropriate computer gaming and social media use on students' ability to accomplish higher academic goals.
2. Discuss the contribution of increased access to digital technologies and social media to cyber bullying practices.
3. What sort of new skills are required to get more job opportunities in the current 21st-century work culture and ethics? What can you do to get the required skillsets?
4. How can students use online resources and materials without plagiarizing?
5. Form a pair of students to discuss how your families and friends use technology. Discuss how your relationships changed because of it and share what you have discussed to class.
6. Discuss with a partner about your habits of technology use and social media usage. What changes are needed to improve your current habits for future?

7. Individual case study: pick a relevant topic that interests you so far from ethical and social issues in information systems, then research and prepare a half page report using any word processing applications. Make sure to acknowledge the sources of information's when presenting information.
8. Conduct online research about known cyber-attack attempts in our country and what is being implemented overcome the risks. Prepare a report by searching the last 2-3 years reports about cyber-attack attempts, common types of attacks and mitigation activities taken to ensure national cyber security.
9. Take a time and think of a type of job that you would like to have in the future. Explore required professional skills, list of potential universities to study the program and other information's about the specific job that you aspire to have.
10. Form a group of 3 students. Find out list of jobs that have been made obsolete by advancement of technology like artificial intelligence, and jobs that are being created as a result of advancement of technology. Present your findings to the class with your justifications

1.1.3 Environmental issues caused by Information Systems

Brainstorming 1.3

- What is the impact of information systems on environmental pollution?

While we make advances in the use of developing technologies including information systems, we should also consider their negative effects on the environment around us.

Technologies consume lots of resources and power which creates additional impacts on our environmental ecosystem. Sometimes people can get so excited about using a new technology, they overlook the negative impact on the environment. Using technologies smartly and responsibly is essential in order not to create problems while utilizing the advantages. Because electronic devices contain hazardous materials that are unsafe for the environment. Most electronic devices

contain non-biodegradable materials, heavy metals, and toxic materials like cadmium, lead, and mercury which can leak into the ground, where they can contaminate the water we drink, the plants we eat, and the animals that live around the area. This clearly shows that is our responsibility to protect the environment while we use and dispose of any information system components. To this end, advanced electronic waste (e - waste) management techniques shall be applied to minimize the damage (See Figure 1.11). e-waste management refers to a systematic procedure for gathering e-waste, recovering and recycling material in a safe manner, and disposing of e-waste in a way that minimizes negative environmental effects.

Figure 1.11: e- Waste management

(Source https://plopdo.com/)

Negative impact of information systems on the environment have also led to the development of new environmentally friendly technology solutions aiming to help solve some of the biggest environmental concerns that we face as a society through a shift towards a more sustainable, low-carbon economy. This includes green or clean technologies which refers to environmentally friendly technology, which monitor, and or reduce the negative impact of technology on the environment (See Figure 1.12). Examples of green technologies includes but not limited to Solar Panel, Electric Vehicles. smart meters, natural gas, wind Turbines etc.

Figure 1.12: Clean Technology

Technological solutions also allow electronic systems and paperless communication like email and online portals and applications to reduce cutting down of trees for paper for a better environmental protection and management (See Figure 1.13). To this end, information systems allow us to have a worldwide virtual laboratory so, that experts

from all fields can share their research, experiences, and ideas to come up with better, and smarter solutions. Not only does this allow people who are far away from one other to work together, but it also reduces the negative carbon emissions impacts that people would normally cause from traveling to meet with one another.

Figure 1.13: Paper based vs paperless system

Activity 1.3

1. Form a group of 3 students and assess the available green technologies in your community and present the advantages of those green technologies for environmental protections.
2. Discusses how students can contribute for the protection of the environment by becoming environment ambassadors while utilizing the advancement of technological solutions

1.1.4 Health Issues caused by information systems

Brainstorming 1.4

- What are the potential impacts of using information systems on our health?

Health information systems are essential for the rapid generation of high-quality data that will be used to inform all relevant stakeholders for program planning, review, and monitoring as well as generally ensuring and enhancing the quality of all components of the health system.

The widespread use of information systems and access to mobile device in healthcare can make it easier to offer care, improve care delivery, empower patients through personalized messaging, and collect real-time data to optimize resources and decision-making. However, we must also pay attention to health-related challenges that caused by unmanaged use of information system services and products on users. Some of the challenges include, sleeping disorders and lack of productivity, loss of attention and stress. At the same time, the current widely used mobile devices to access health information systems and other online services also creating computer related health problems that needs a serious attention by all users.

Examples of such problems incudes but not limited to musculoskeletal problems on our body such as our back, neck, chest, arms, shoulders and feet, Vision Problems and eye strains, Headaches, Obesity due to long hour of sitting etc..

This implies users should be aware of the proper ways when using information systems and associated access devices to get the expected health information system benefits by minimizing the harm.

Figure 1.14: Health Information System

Activity 1.4

- Discuss the roles of availability of mobile devices for accessing health-related information systems services.

1.2. Intellectual Property

Brainstorming 1.5

- What comes to your mind when you hear the term Intellectual Property

Intellectual property rights refers to a range of intangible rights of ownership of an asset such as a software program or designed information system.

Intellectual property rights provides different methods for protecting these rights of ownership based on their type. As stated by Ethiopian Intellectual Property Office Establishment Proclamation, Proclamation No. 320/2003 "Intellectual property" means a legal right over a creative work of the human intellect and includes patent, trademark, registration certificate and copyright".

There are essentially four types of intellectual property rights that are relevant to the software and information system industry. These are :-trade secrets , patents, copyrights, and trademarks. We used World intellectual property organization (WIPO) definitions:

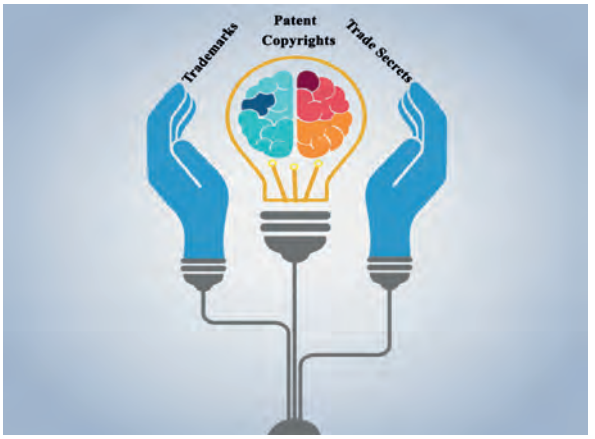


Fig 1.15: Intellectual Property Right

- **Trade secrets:** "*Trade secrets are Intellectual Property rights on confidential information which may be sold or licensed. The unauthorized acquisition, use or disclosure of such secret information in a manner contrary to honest commercial practices by others is regarded as an unfair practice and a violation of the trade secret protection*".

- Examples of trade secrets include customer lists, source code, recipes for certain food items and beverages including Pepsi, Coca-Cola, and Sprite), software processes, and more.

- **Patent:** *"A patent is an exclusive right granted for an invention. Generally speaking, a patent provides the patent owner with the right to decide how - or whether - the invention can be used by others. In exchange for this right, the patent owner makes technical information about the invention publicly available in the published patent document''*.

  When a property owner holds a patent, others are prevented, under law, from offering for sale and making or using the product.

  Examples of patented products include software programs like Airbnb, Amazon, One-click, the page rank algorithm of Google, and other inventions including the light bulb, telephones, Bluetooth, and more. Not all software products can have a patent. The software has to meet certain standards to get a patent. These standards come from the responsible authority in each country.

  In our country, the Ethiopian Intellectual Property Office is the legal entity to manage Intellectual property related applications. There are defined set of application procedures to follow in order to apply for patent. The application can be processed online through the application portal.

  You can get detailed information, application and registration procedures from its official website (https://eipo.gov.et/) for all intellectual property right applications and questions.

Fig 1.16: Ethiopian Intellectual property Authority Website

(Accessed on 27/04/2022 G.C, 2:00 P.M EAT)

- **Copyrights:** Copyrights and patents are not the same although people are often confused. A copyright is a type of intellectual property protection that protects original works of authorship, which might include literary works, music, art, and more. Nowadays, copyrights also protect computer software and architecture. Copyright protections are automatic; once you create something, it is yours. However, if your rights under copyright protections are infringed, and you wish to file a lawsuit, then registration of your copyright will be necessary. Graphical User Interfaces (GUIs) can also be registered for copyright.

- **Trademarks:** *"A trademark is a sign capable of distinguishing the goods or services of one enterprise from those of other enterprises"*.

  Example- 'Meta','RIDE' and 'Microsoft' symbol–which nearly all could easily recognize and identify–are types of trademarks. There are also many companies in our country with registered trademarks. Examples are Ride, Meta, and Microsoft. While

Fig 1.17: Trademarks of Meta, Microsoft and Ride

patents and copyrights can expire, trademark rights come from the use of the trademark, which can be held indefinitely. Like copyright, registration of a trademark is not required, but registering can offer additional advantages.

Activity 1.5

1. What are the differences between a patent and a trademark?
2. Koket Destalem Ayde, a grade 12 student, invented a unique software product that can automate the existing health system in our country. Koket is not aware of whether she can have a patent right for her product or not. Refer to Ethiopian Intellectual Property Authority's official website and other relevant resources to come up with the requirements to get a patent and the application process to follow . Form a group of 3 students and find all relevant information, prepare a report and present it to the class. Whether Koket's software product can have patent right or not.

1.3. Digital Identity Management

Brainstorming 1.6

- What makes our digital or online identities different from our physical identities?

As stated by onfido.com, "a digital identity is a collection of information about a person that exists online". This data can be used to create a digital image of a person when it is collected. Such information can be used by businesses to identify their clients.

A digital identity is all of information about an individual, organization, or electronic device that exists online. Every individual should have the right to a trusted and secured digital identity to access fundamental services, like financial services, mobile connectivity, health,

Figure 1.18: Digital Identity Attendance controlling device using fingerprint

education, and other services.

For almost every organization, digital identity management presents both significant opportunities and associated risks. Management of a digital identity involves limiting the data that a person gives out, avoiding excessive use of storage, checking what data is stored about each individual, making corrections to incorrect personal data, avoiding redundant data, etc.

Digital identification can be verified in a variety of ways, including through database checks, document and identity verification, biometric verification, and identity verification. A user can access internet services including banking, health, and travel after proving their identity.

There is an ongoing effort at a national level to enhance citizens' digital identity management practices. The National ID project started as part of the Digital Ethiopia 2025 strategy in Ethiopia. It is a robust and reliable digital identification process for all Ethiopians and legal residents within the country.

Figure 1.19 Ethiopia national ID project

Activity 1.6

1. How can users manage their digital identity in the digital space to minimize security concerns?
2. What advantages can be obtained from current national digital identity management initiatives like National ID expand the quality and access of digital services in the country?

1.4. Collaboration Through Digital Technologies

Brainstorming 1.7

- What is digital collaboration and how it can change the way we learn?

Digital collaboration means working with other individuals or teams and getting things done using digital devices and technology. It can include different tools, apps, software, procedures and processes. Digital collaboration uses technology to improve workplace collaboration, communication, document management, content management, and information flow across the organization which in turn leads to improved employee productivity and efficiency. Digital collaboration has become important and is relevant to most organizations, groups, communities, schools and creative groups and in the current digital world.

To this end, Internet and Internet-based applications is an important part of digital collaboration as they helps employees collaborate in real-time, even beyond time zones and geographical boundaries.

Figure 1.20: Collaboration through digital technologies

Links

See Cloud and Cloud-based applications in detail in Unit 2 Section 2.3 of this textbook for more details.

1.4.1 Components of Digital Collaboration

People: People are at the heart of collaboration. Developing workflows and established practices ensure they can stay in touch, especially when working remotely due to work schedule preferences or pandemic restrictions like COVID 19. Proper arrangement and communications is required while working on similar projects and assignments to remove unnecessary work or duplication of efforts in collaborative working environments.

Collaborative tools: There are many tools and systems available that foster collaboration, but not every solution will be a good fit due to its features, prices, usage polices and ease of use. The choice of technology, along with how well employees use it will determine effectiveness of the collaboration. Digital Collaboration tools includes many types including the following with their examples:

- **Communication tools**: Communication tools include options for integrated messaging, co-working services, Privacy and access settings, Conference calling, Video chats with screen sharing etc.
    - Examples- Microsoft Teams, Skype, Google Meet, Zoom, Cisco Webex, Slack, Microsoft Yammer, and more
- **Documentation and file sharing tools**: This tools includes features like Support for different file types, Comments, and notes attached to files, track changes and versioning, editing permissions and access control, folder sharing , and more.
    - Examples- Google Drive, Google Docs, Dropbox, MS SharePoint, Media Fire, and more.
- **Project Management tools:** It has underlying project and task management capabilities at their core.
    - Examples- Trello, Asena, ClickUp, and more.
- **Data Visualization tools:** These tools should support a range of visual styles, be easy and quick to use, and be able to handle a significant amount of data.
    - Examples- Google Charts, Tableau, Power BI, Grafana, Chartist, FusionCharts, Datawrapper, Infogram and more.
- **Notetaking tools:** These tools help to quickly capture and organize your notes, ideas, web page content and more.
    - Examples- Evernote, Microsoft OneNote and Google Keep and more.

Device: Devices are also key players. Devices such as smartphones, tablets, and laptops are often an afterthought, but they are important components of digital collaboration. They are the physical media we use to connect.

Activity 1.7

- Describe and explain collaboration tools you. Use at schools or in a relevant groups.

Activity 1.8

This is a group practical activity. Form a group of 5 students from your class and pick a name for your group. Follow the instructions below to complete an activity using a digital collaboration tools.

Step 1- Prepare a half-page summary about the benefits of digital collaboration tools for online learning and group assignments and projects. Then select cloud-based word processing software and create an online word document and name it "Digital Collaboration Activity _ Your group name" and put your half-page summary on the newly created online word processing file.

Step 2 - Group members create the online document and send the link to all members via email or preferred social media platforms

Step 3 - Select your chosen screen-sharing collaborative tools for virtual meetings (Zoom, Google Meet, Microsoft Team, or any other application).

Step 4 - Group member to generate a meeting link, and another to join the virtual meeting to debate and review the summary document. Use the selected tool's screen-sharing functionality and change the document from wherever you are.

Step 5 - Discuss how collaborative tools like these can improve learning and teamwork activities, and share your experience with your classmates.

1.5. Engaging in Citizenship through Digital Technologies

Brainstorming 1.7

- How can you apply digital technologies to develop citizenship engagement?

According to aeseducation.com , digital citizenship refers to the responsible use of technology by anyone who uses computers, the Internet, and digital devices to engage with society on any level.

It require the ability to engage positively, critically, and competently in the digital environment to practice forms of social participation that are respectful of human rights and dignity through responsible use of technology.

Young people needs to be supported to participate safely, effectively, critically, and responsibly in a world, which is filled with social media and digital technologies.

Good digital citizenship teaches peoples how to use digital tools to communicate with one another, empathize with one another, and build long-lasting connections. On the other hand, poor digital citizenship includes things like online bullying, careless social media use, and a general ignorance of Internet safety.

That is why digital citizenship should not be a long list of don'ts. It should be about the do's that help create thoughtful, empathetic digital citizens who can struggle with the important ethical questions at the intersection of technology and humanity.

The do's include:

- ✓ Using technology to make their community better.
- ✓ Engaging respectfully online with people who have different beliefs.
- ✓ Using technology to make their voice heard by public leaders and the global community.
- ✓ Determining the validity of online sources of information.

Digital citizenship goes beyond conversations about personal responsibility. It is about being active citizens who see possibilities instead of problems, and opportunities instead of risks as they curate a positive and effective digital footprint.

Example – In 2021 G.C Ethiopians united for the Grand Ethiopian Renaissance Dam (GERD),"#It_is_my_dam" social media Hashtag movements using Twitter, Facebook, Instagram, and LinkedIn to express their full support for our national grand project and to create awareness about the purpose and intention of the project for the global community during the process of dam filling.

#ItsMyDam

Figure 1.21: Grand Ethiopian Renaissance Dam

Activity 1.9

1. Discuss possible ways that students can raise their voices about high-priority national agendas like the Grand Ethiopian Renaissance Dam as part of a true digital citizenship movement to raise awareness among the global community and policymakers.
2. Prioritize and select a community agenda items that needs an awareness creation movement. Then, use digital tools and social media to create campaigns as the community member.

Unit Summary

In this unit, you have learnt about:

- ethical and legal issues in information systems
- issues related to the social implications of information systems.
- issues related to environmental and health implications of Information systems.
- intellectual property rights and types of intellectual property rights.
- digital identity management
- how to facilitate collaboration through digital technologies
- how digital collaboration tools can be applied in our daily lives
- how you can engage in citizenship through digital technologies.

Key Terms

Copyright - A copyright is a type of intellectual property protection that protects original works of authorship, which might include literary works, music, art, and more

Digital citizenship- refers to the responsible use of technology by anyone who uses computers, the Internet, and digital devices to engage with society on any level.

Digital collaboration - means working together and accomplishing goals while using digital tools and technology. It may consist of anything, such as devices, programs, software, steps, or an entirely new technique.

Digital identity is an online or networked identity adopted or claimed in cyberspace by an individual, organization, or electronic device.

Intellectual property right - refers to a range of intangible rights of ownership as an asset such as a software program or designed information system

Information Systems – are interrelated components working together to collect, process, store, and disseminate information to support decision-making.

Information Technology (IT) - is the use of any computers, storage, networking, and other physical devices, infrastructure, and processes to create, process, store, secure, and exchange all forms of electronic data

Patent - a patent is a type of limited-duration protection that can be used to protect inventions (or discoveries) that are new, non-obvious, and useful, such as a new process, machine, article of manufacture, or composition of matter.

Trade secret- refers to specific, private information that is important to a business because it gives the business a competitive advantage in its marketplace

Trademarks refer to phrases, words, or symbols that distinguish the source of a product or service of one party from another.

Social media addiction is a behavioral addiction characterized by excessive concern for social media, an insatiable want to access or use social media, and a commitment of so much time and energy to social media that it interferes with other crucial facets of life.

Review Questions

Part I: Write True if the statment is correct and False if it is incorrect.

1. The concern and questions of ethics and legality are essential in many industries including education.
2. Most people have their data spread throughout the digital world which can be accessed by unintended sources.
3. Since information technology becomes increasingly influential, ethical and legal considerations are non-relevant and less important.
4. In the current digital world, security issues have been resolved by locking a door only.
5. To fulfill the objectives of environmental management, technological advancements require systems that can make proper and ethical of information systems and telecommunication infrastructures.
6. Most people have complex online lives, and they may not even realized how their personal information is being collected and used.

Part II: Choose the correct answer from the given alternatives.

1. Which one of the following is NOT among four pressing ethical and legal issues confronting the industry today.

   A. Data Creation
   B. Privacy
   C. Data gathering
   D. Digital ownership

2. What types of data can be collected from customers' purchase behavior by companies even without their consent in some cases?

   A. Purchasing patterns,
   B. Personal preferences,
   C. Professional/social affiliations
   D. None

3. Which type of information system structure is seen in the environment in which people collaborate to deal with environmental problems?

   A. Environmental Information systems
   B. Health Information System
   C. Agricultural Information System
   D. Enterprise Resource Planning System

4. Health-related negative effects of information system include all except_____.

   A. Adaptability to use technology
   B. Loss of attention and stress
   C. Sleeping disorders
   D. Internet addiction

5. Which one used to protecting inventions, discoveries, expressions of ideas, and creations, like art and writing?

   A. Patent
   B. Trademark
   C. Trade secret
   D. Copyright

Part III: Discussion Questions.

1. What will be the possible legal consequences of violating ethical issues in our information systems use?
2. Discuss the emerging health issues associated with the current digital devices use and systems applications in our day-to-day life
3. What are the 3 common components of a digital collaborations? Discuses each component meaning and roles for effective digital collaboration engagements.
4. What is the role of technology in enhancing digital citizenship?

Part IV: Group Case Study

1. Explore and select two schools, higher institutions, learning academies, or private business in your community that are implementing digital collaboration tools for their daily operations. Prepare a case summary document by including
   - the type of the organization you choose and its vision
   - the types of digital collaboration tools the organizations/schools/ institutions are using in their working environments.
   - major educational or business functions supported by those digital collaboration tools.
   - advantages and limitations encountered while using the selected digital collaboration tools.

UNIT 2

EMERGING TECHNOLOGIES

Learning Outcomes

At the end of this unit, students will be able to:

- Describe the meaning and use of Bigdata
- Explain cloud computing and fog computing
- Apply cloud computing services
- Explain the Internet of Things (IoT) and its applications

Unit Overview

Emerging Technology is a term that is commonly used to describe a new technology, but it can also refer to the evolution of existing technology. It is widely used to describe technologies that are in development or will be available within the next five to ten years. It is usually reserved for technologies that are creating or are expected to create significant social or economic effects. When employed in different contexts, such as media, business, science, and education, it might have slightly different meanings. You have learned selected emerging technology concepts including data science, artificial intelligence (AI), virtual reality, and augmented reality in your grade 11 Information technology (IT) course. This unit addresses some additional selected emerging technologies including Bigdata, cloud computing, fog computing, and internet of things (IoT).

2.1. Introduction to Bigdata

Brainstorming 2.1

- What is Bigdata? What makes it different from data?

We are living in a digital world where a tremendous amount of data is generated by each digital service we use. The amount of data generated is increasing in different dimensions including data sizes. This vast amount of generated data is leading us to the creation of Bigdata. Bigdata refers to a collection of data sources that are so massive and complex that they become challenging to process using typical data processing software or readily available database management tools (See Figure 2.1).

Figure 2.1. Concept of Bigdata

Bigdata starts with the exponential explosion in the amount of data we have generated since the dawn of the digital age. This is largely due to the rise of computers, the Internet, and technology capable of capturing information from the real and physical world we live in, and converting it to digital data that can benefit businesses and organizations (See Figure 2.2).

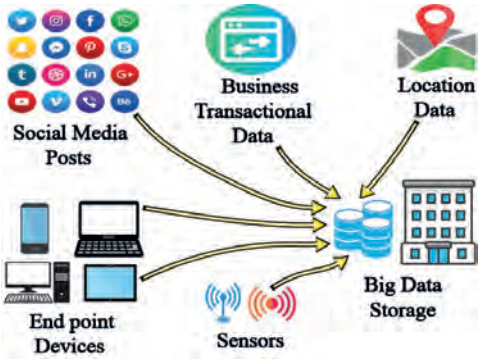


Figure 2.2. Bigdata source to storage

In Bigdata collection, the range of organizational sources generating data needs to be identified. These sources include business firms producing transactional data such as point-of-sale software, endpoint devices within IoT ecosystems and marketing firms, Social media posts from existing and prospective customers, multiple additional sources like smartphone locational data, surveys that directly ask customers for information are also some of the sources of Bigdata. Identifying useful data sources is the start of the Bigdata collection process. From there, an organization must build a pipeline

that moves data from generation to enterprise locations where the data will be stored for organizational use.

Example – Global companies like Google use Bigdata to understand what users want from it based on several parameters such as search history, locations, trends, and many more.

2.1.1 Characteristics of Bigdata

Bigdata is characterized by 5V, namely Volume, Variety, Velocity, Veracity, and Value as stated by tectarget.com. It should also be noted that characteristics of Bigdata are subjected to additional features as scientific research and trend continues in Bigdata analytics. The most common characteristics of Bigdata are discussed by considering the 5 V's (Figure 2.3).

Figure 2.3. 5V's Characteristics of Bigdata

- **Variety:** Variety of Bigdata refers to the nature of data i.e whether it is structured, unstructured, and semi-structured data. In the past, data could only be collected from spreadsheets and database. Today, data comes in an array of forms such as emails, PDFs, photos, videos, audio, social media posts, and others. Variety is one of the important characteristics of Bigdata (See Figure 2.4).

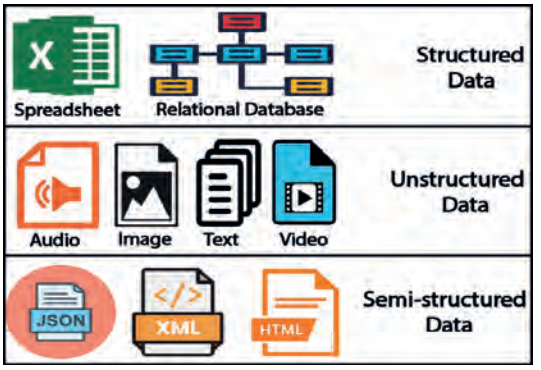


Figure 2.4. Variety in Bigdata

- **Velocity:** Velocity refers to the speed at which data is being created in real-time.
- **Volume:** Volume indicates huge 'volumes' of data that are being generated daily from various sources, like social media platforms, business processes, machines, networks, human interactions, and others.

- **Value:** Value is the other major issue that we need to concentrate on. It is not just the amount of data that we store or process. It is the amount of valuable, reliable, and trustworthy data that needs to be stored, processed, and analyzed to find insights.

- **Veracity :**Veracity means the degree of reliability that the data has to offer. Bigdata focuses to find an alternatives way to filter the huge unstructured data to use in business developments.

Links

Unit 2 of Grade 11 IT Textbook Introduction to Emerging Technologies about structured, unstructured, and semi-structured types of data.

2.1.2 Benefits of Bigdata

Brainstorming 2.2

- What benefits would a company get from analyzing Bigdata?

Bigdata projects help us to cure disease and prevent cancer, maximize crop yields, explore distant planets, predict and respond to natural and man-made disasters, prevent crimes, and more. There are plenty of project initiatives in Ethiopia to benefit from Bigdata. Some institutions taking the initiative to use Bigdata to maximize customer services and properly segment customers data and among the initiatives Ethiopian Airlines, Ethio Telecom, Artificial Intelligence Center, and some banks started leverage its benefits

Bigdata has many advantages. Some of them are discussed as follow.

- **Customer acquisition and retention**: Consumer data can help the marketing efforts of companies, which can act on trends to increase customer satisfaction. For example, personalization engines for Amazon, Netflix, and Spotify can provide improved customer experiences and create customer loyalty.
- **Targeted advertisements (Ads):** Personalization data from sources such as past purchases, interaction patterns, and product page viewing histories can help generate compelling targeted advertising campaigns for users both on the individual level and on a larger scale.

- **Product development**: it allows you to update existing products/services while innovating new ones..
- **Price optimization:** This minimizes the manual work and reduces the possibility of any man-made errors.
- **Risk management**: Bigdata analytics can identify new risks from data patterns for effective risk management strategies.
- **Improved decision-making:** Insights of business users that are extracted from relevant data can help organizations make quicker and better decisions.

2.1.3 Application of Bigdata

Below are some examples of sectors in which Bigdata can contribute by generating value:

- **Healthcare** - In the healthcare industry, Bigdata has already begun to make a significant impact. Healthcare analysts use Bigdata to inform health prevention, intervention, and management using a variety of data analytic technologies and approaches. These efforts can improve the patient experience, care efficiency, and quality, and reduce healthcare costs (See Figure 2.5).

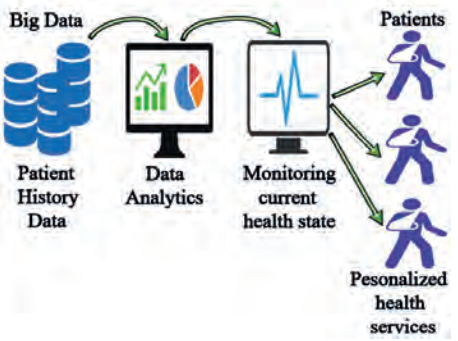


Figure 2.5. Health application of Bigdata

Patient medical and hospital records, medical exam findings, and information acquired by healthcare testing machines are all examples of Bigdata sources in health care. Apart from that, Bigdata and AI-powered fitness devices, telemedicine, and remote monitoring are all helping to improve lives.

- **Education** - Bigdata can also be utilized to improve education by providing students with a balanced learning plan. Educators can tailor programs for learners based on their grades and attention spans. Students may also be offered blended learning, which combines

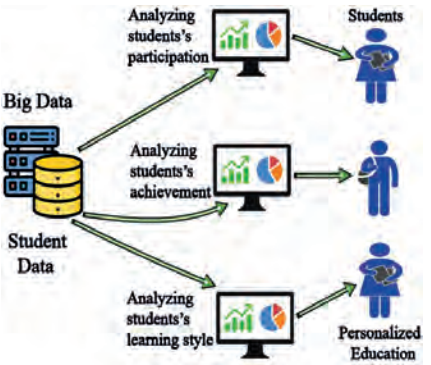


Figure 2.6. Education application of Bigdata

offline and online learning possibilities (See Figure 2.6).

Bigdata is being used by educational institutions to evaluate applications and determine who will be a good fit for the institution and who will not. This has assisted institutions all around the world in shortening the selection process which creates different learning possibilities. Bigdata can improve student experience, academic programming, more effective evidence-based decision-making, and a strategic reaction to shifting global trends.

- **Banking** - Bigdata solutions can detect fraudulent behaviors in real-time, such as credit/debit card usage, inspection track archiving, and more. Bigdata analysis also helps banks in their compliance verification, auditing, and reporting processes. This simplifies the processes while lowering overhead costs. Nowadays, people provide comments on a financial institution's work by phone or on the website, and they share their thoughts on social media (See Figure 2.7).

Figure 2.7. Banking application of Bigdata

- **Agriculture** - Smart farming and precision agriculture practices help farmers to save costs and open new business opportunities. Granular information on rainfall patterns, water cycles, fertilizer needs, and other topics is made available to farmers through big data. They can use this information to make wise choices about when to harvest and what crops to grow for maximum profit. Making the proper choices ultimately increases agricultural outputs.

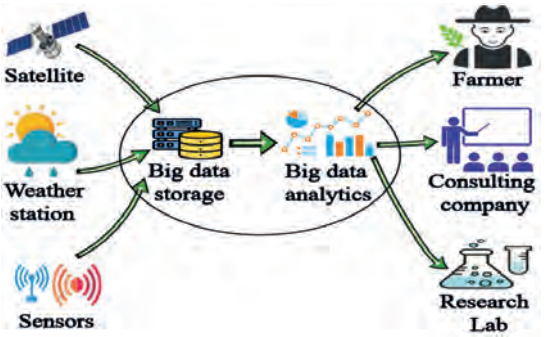


Figure 2.8. Agriculture application of Bigdata

- **Manufacturing** - In the manufacturing sector, Bigdata helps create a transparent infrastructure, predicting uncertainties and incompetence that can affect the business adversely (See Figure 2.9).

Figure 2.9. Manufacturing application of Bigdata

- **Retail** - Bigdata has changed the way traditional retail industry. works. Over the years, retailers have collected a vast amount of data from local demographic surveys, Point of Sale (PoS), scanners, customer cards, and store inventory. Now they have started to use this data to create personalized customer experiences, boost sales, increase revenue, and deliver improved customer service (See Figure 2.10).

- **Transportation** - The transportation business can also benefit greatly from Bigdata analytics. Both companies and government-run transportation corporations employ Bigdata technologies to optimize route planning, control traffic, manage road congestion, and improve services in countries all over the world. We also use Bigdata in other industries, such as media and entertainment, energy and utilities, government-related activities, and others (See Figure 2.11).

Figure 2.10. Retail application of Bigdata

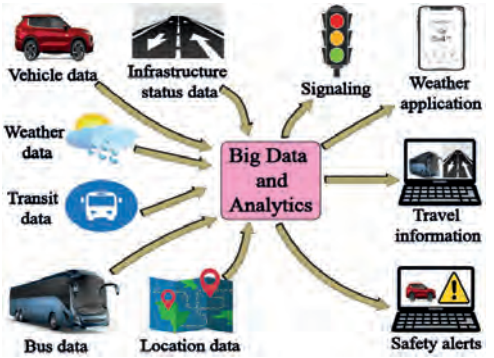


Figure 2.11. Transportation application of Bigdata

Activity 2.1

- All global companies, including Facebook, Twitter, Amazon, Alibaba, and other global leading companies, have turned to Bigdata as a powerful tool for monitoring their customers' social media engagements. This help companies to understand how their customers talk about brand experiences by analyzing large volumes of customer experiences feedbacks and suggestions.

  1. What kind of consumer data will be collected to improve organizations' service delivery and customer management?
  2. How can businesses analyze and use the data they collect from customers?
  3. What are the potential drawbacks of using Bigdata technologies and methodologies to analyze customer data?

2.1.4 Challenges of Bigdata

Aside from the benefits, Bigdata also has challenges related to data quality, storage, a shortage of data science experts, validating data, and gathering data from various sources. Even though the challenges of Bigdata and other similar emerging technologies will change over time due to the dynamic and frequent advancement of those technologies, below are some of the current major Bigdata challenges.

Managing Bigdata Growth- One of the main issues with Bigdata is how to properly store these large amounts of data. Data centers and databases used by businesses are constantly storing more data. Bigdata sets grow quickly over time, making management of them more difficult

Lack of data professionals- Companies demand skilled data specialists to manage Bigdata solutions. These experts consist of data scientists, data analysts, and data engineers who are skilled at using the tools to manage huge data sets. Students who are passionate about data science disciplines can think of studying and practicing concepts related to data science and Bigdata analytics as part of their career plan.

Securing Data - One of the major difficulties of Bigdata is keeping these enormous sets of data secure. Companies frequently put data security to later phases because they are so busy in understanding, storing, and analyzing their data sets. Attentions is required to protect data repositories unless it serve as a haven for intruders.

Integrating data from a variety of sources- Data is obtained from different sources in a business and operations. This includes social media pages, company wide software products, customer log files, financial transaction data, email communications, presentations, and reports by employees. It require to determine which data is most relevant and focus on that to save organization time and money.

2.2. Cloud Computing

Brainstorming 2.2

- Why do you think the name' cloud' is used for cloud computing?

2.2.1 Introduction to Cloud Computing

Cloud computing is defined as storing and accessing data and computing services over the Internet. It does not store any data on your personal computer. Computing services like servers, data storage, networking, databases, etc. are all available on-demand. The main advantage of cloud computing is it gives access to many users from single data center which allows users to store and access their data over the internet without any limitations.

We have all experienced cloud computing at some point. The goal of cloud computing is to provide easy and flexible access to computing resources and IT services, various which hardware and software components are required for the proper use of a cloud computing model.

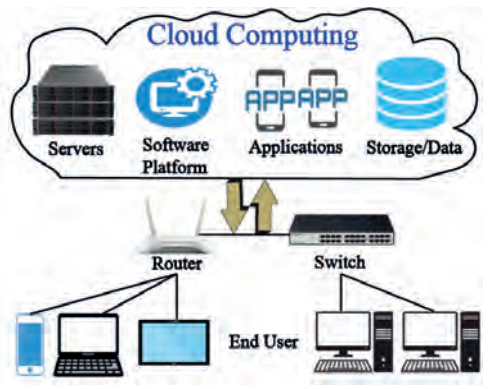


Figure 2.12. Cloud Computing Basics

Peoples are using different types of cloud-based services for their email communications, document management, entertainment and ecommerce services. Examples of cloud-based services includes but not limited to: Gmail, Google Maps, Amazon, Netflix, PayPal, Spotify, Adobe Creative Cloud, Kindle, etc.

2.2.2 Benefits of Cloud Computing

Accessibility and mobility- Users may access their tools from anywhere, at any time, and from any type of devices as long as they are connected to the Internet, which is one of the most significant benefits of the Cloud.

The ease of access to data in the cloud enables users to rapidly store and retrieve resources from distant servers, work from any location with an internet connection, and communicate with peers without being physically present in the same space. Users do not lose any data if their services fail because the Cloud service provider backs up their data (See Figure2.13).

This greatly helps the implementation of inclusive education which is schools for everybody through support learning and respond to individual needs. Cloud based services support to include inclusive technologies like voice recognition, mind mapping, word prediction, text to speech for editing etc. to accommodate special needs learners.

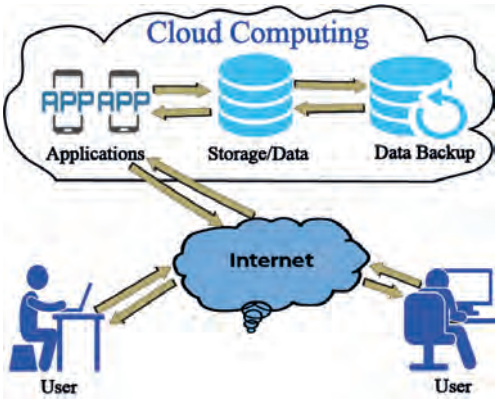


Figure 2.13. Accessibility and mobility

Flexibility- Services become entirely adaptable and can be changed at any time to meet users' needs and commercial requirements. User can restrict or enhance the available resources while paying only for what is used. Another benefit is that resources can be shared, allowing users to collaborate in real-time on the same content in all domain and work environments (See Figure 2.14).

Updating and scalability – The service provider, as well as taking care of maintenance, takes care of all service updates, which allow users to focus more effectively on our work and responsibilities while increasing our efficiency.

Optimized security- When compared to traditional and internal infrastructures, Cloud Computing provides enhanced security. Companies are finding it increasingly difficult to effectively secure their own IT systems due to a lack of time, expertise, and budget (Figure 2.15).

Controlled costs- The other advantage is that Cloud Computing allows for greater control of costs IT infrastructure. We can adjust the resources allocated to its infrastructure depending on its needs by increasing or decreasing the resources available. By subscribing to Cloud Computing services, users or companies pay only for what they used (Pay as you go model). This helps users to no longer deal with costs relating to maintenance and renew computer equipment (servers, computers, etc.) Companies and users can reduce their IT investments and optimize their operational and business costs (Figure 2.16)

Example- Let's think that you plan to publish your designed website activity.

Now you want to host your website and make it public. Thus, here you have two possible options

Option 1- Buying physical server and hosting it in your data room by getting a public Internet Protocol (IP) address

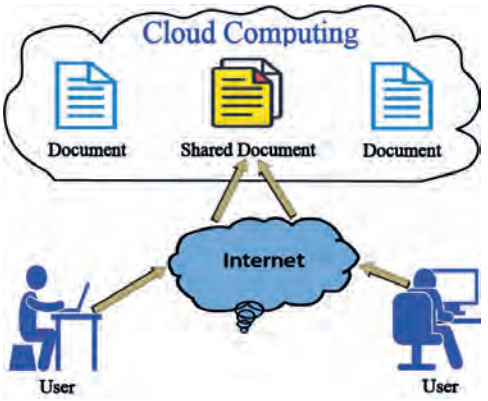


Figure 2.14. Flexibility

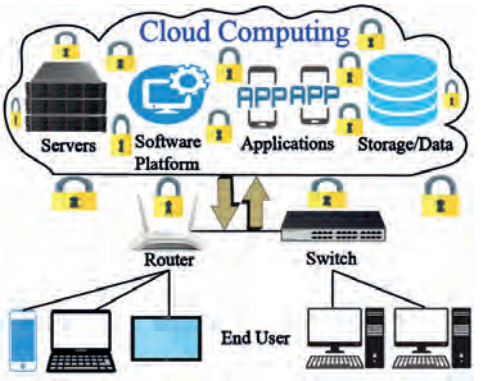


Figure 2.15. Optimized Security

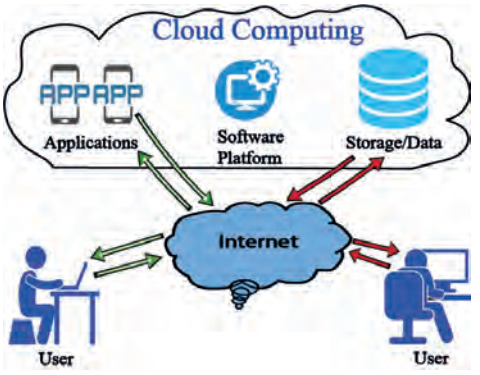


Figure 2.16. Controlled Cost different users getting different services

from your service provider.

LINK - See Unit 4 of Grade 9 and Grade 10, IT textbooks about computer networks and IP addresses for more details.

Then, you need to install an operating system on the server and web services to run your website. Now you are ready to publish your website, but the issue is that you have already invested a lot of time and money in getting your site available on the Internet. Right from buying server hardware to hosting it in a data center, the process costs you lots of money and time. This was a common situation a decade ago (See Figure 2.17). Now we have another alternative, a Cloud-based service to host our websites.

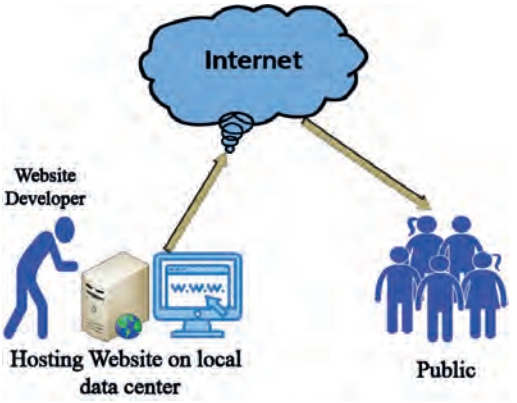


Figure 2.17. Publishing website without cloud technology

Option 2. In Cloud-based services like website hosting, you do not need to buy new server hardware of your own rather you can borrow it for any time as long as you get at less price, which can save both money and time. Secondly, you do not have to manage operating systems and web services on your own, rather a cloud service provider manages them. You get a button to upload your website content, and once data is uploaded, your website content is available to the public. This process costs you much less money and gets ready in a few minutes (See Figure 2.18). This can be a simple example to understand the benefits of cloud computing. The benefits of cloud services are applicable to different services which are covered in the next section.

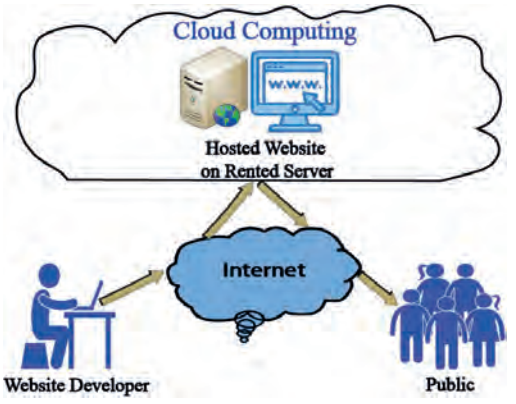


Figure 2.18. Publishing website on Cloud technology

Activity 2.2

1. Mention at least 2 cloud services you have used before for your academic and social activities. Explain how you started to use it and what benefits you get from the services.
2. Download and Install Cloud based file-sharing application. Then upload documents and practice file sharing among your friends or families

2.2.3 Limitations of Cloud Computing

Cloud computing are not without their limitations. However, It's critical to weigh both the benefits and limitations of a new service before choosing if it is the best fit for us. Below are some of the limitations of cloud computing:

- **Server downtime** – No cloud provider can guarantee that you will not have service interruptions, which will prevent you from accessing your data or files. Because cloud computing is dependent on an Internet connection, if it goes down, so will your cloud services.
- **Security and privacy issues** – When it comes to handling sensitive data, companies are always concerned about how safe and private it is. There is still a risk of intruders, even though a cloud provider would have security processes in place to ensure that your data is maintained and protected properly. Intruders are constantly attempting to find new ways to break into cloud providers' systems in today's digital world. Therefore, cloud computing services are not completely safe.
- **Data ownership and transparency** – Companies may think that there is a lack of transparency concerning data ownership. Does data that is uploaded to a cloud computing system become the cloud provider's property? Another concern is lack of data openness. Companies may be concerned about how their data is managed and whether it is being compromised or not.
- **Inflexibility** – Some cloud computing providers may take a corporation off guard by requiring the usage of their applications and formats. For example, they can limit that a corporation can only upload data developed

in a specific application program. Because the cloud provider has complete control over your cloud service infrastructure, they have ultimate authority over how your data is handled.

- **Lack of support** – Cloud-based services, in comparison to other hosting services and providers, provide minimal to no customer assistance. Customers who are unable to contact cloud providers by phone or email frequently turn to Internet forums for answers to their questions.
- **Cost** – Another disadvantage of this service is the unknown fees. Although cloud hosting eliminates the need for on-site infrastructure, it is often necessary to ensure that you are paying for all of the features that your organization needs to get the most out of the service. A cloud computing service can become costly in this regard.

2.2.4 Types of Cloud Computing

There are mainly four types of cloud computing available currently, namely:

1. public cloud
2. private cloud
3. hybrid cloud
4. community cloud

Public cloud– It is a standard cloud computing model where a service provider's resources are shared across different clients (See Figure 2.19). For example, if you rent 10GB of public cloud storage, you will get the requested capacity of a disk from a bigger chunk of a single storage device that is logically divided into small parts and distributed to multiple users isolated to make it private across different clients.

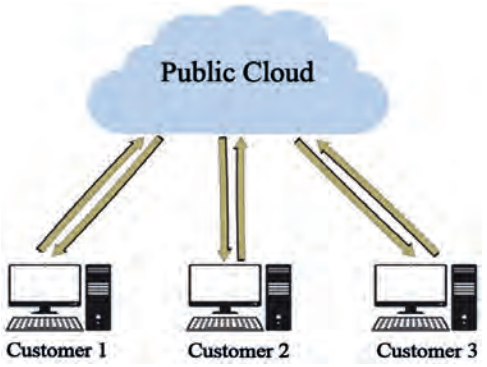


Figure 2.19. Public cloud

Private cloud – As the name indicates, the private cloud has all resources dedicated to your application. It can be either from a cloud service provider or your company's data center. It gives complete control over data security and privacy. The private cloud provides an additional layer of protection to process sensitive and confidential information (Figure2.20).

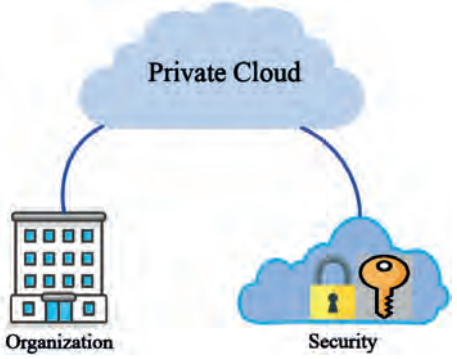


Figure 2.20. Private cloud

Hybrid cloud – It is the combination of public cloud and private cloud. You can host your public service in the public cloud, and confidential and data-related services in the private cloud. It offers a very flexible and comprehensive benefit of both models (See Figure 2.21).

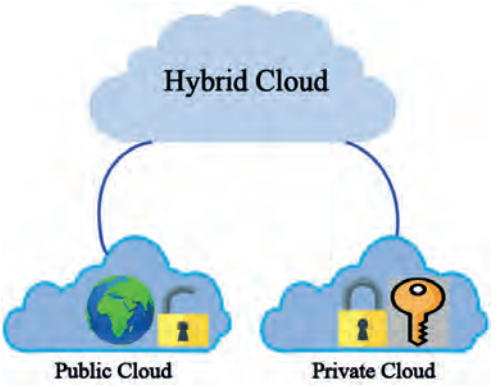


Figure 2.21. Hybrid cloud

Community cloud – It is a collaborative, multi-tenant platform used by several distinct organizations to share the same applications. The users are typically operating within the same industry or field and share common concerns in terms of security, compliance, and performance (See Figure2.22).

Figure 2.22. Community cloud

In essence, a community cloud is a private cloud that functions much like a public cloud. The platform itself is managed privately, either in a data center or on-premises. Governmental institutions, healthcare facilities, financial services companies, and other professional communities frequently use this solutions.

2.2.5 Cloud Computing Services

There are several cloud computing services. The three major cloud computing services are:

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

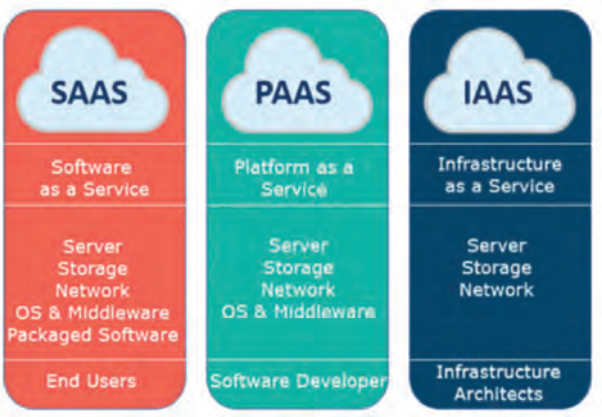


Figure 2.23. Cloud Computing Services

Different businesses use some or all of these components according to their requirements.

Software as a Service (SaaS) - SaaS is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network or the Internet. SaaS is becoming an increasingly prevalent delivery model as underlying technologies supports of web services. Through the Internet, this service is available to users anywhere in the world. Traditionally, software applications needed to be purchased upfront and then installed onto your computer. SaaS users on the other hand, , subscribe to it, usually on a monthly or yearly basis using the Internet.

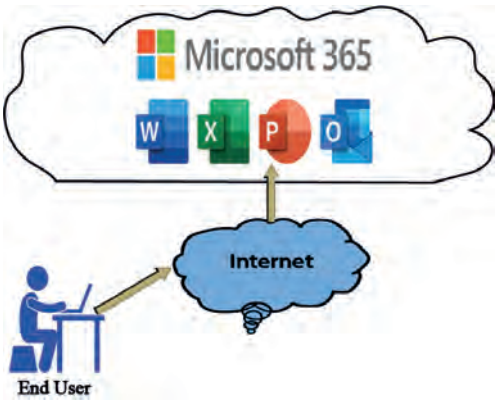


Figure 2.24. Hybrid cloud

SaaS is compatible with all Internet-enabled devices. Many important tasks like accounting, sales, invoicing, and planning can be performed using SaaS (Figure 2.24).

- **Examples of SaaS -** are Google Workspace, Dropbox, Microsoft Office 365, Slack, Cisco WebEx, Concur, GoToMeeting, and many more.

Platform as a Service (PaaS) - provides a platform and environment to allow developers to build applications and services. This service is hosted in the cloud and accessed by the users via the Internet. To understand this, let us compare it with painting a picture, where you are provided with paint colors, different painting brushes, and paper by your school teacher, and you just have to draw a beautiful picture using those tools. Similarly, PaaS provides a platform to support application development (See Figure 2.25). It includes software support and management services, storage, networking, deploying, testing, collaborating, hosting, and maintaining applications.

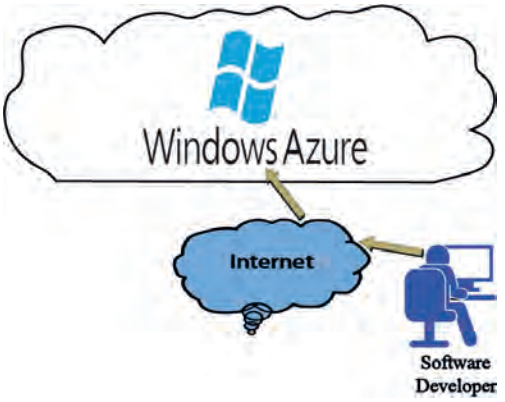


Figure 2.25. Platform as a Service (PaaS)

- **Examples of PaaS** – are Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos, OpenShift, and many more.

Infrastructure as a Service (IaaS)- IaaS is one of the fundamental service models of cloud computing, alongside PaaS. It provides access to computing resources in a virtualized environment- "the cloud"- on the Internet (See Figure 2.26). IaaS is a complete package for computing. For small-scale businesses that are looking for cutting costs on IT infrastructure, IaaS is one of the solutions.

- **Examples of IaaS are -** Amazon Web Services (AWS), Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE), and many more.

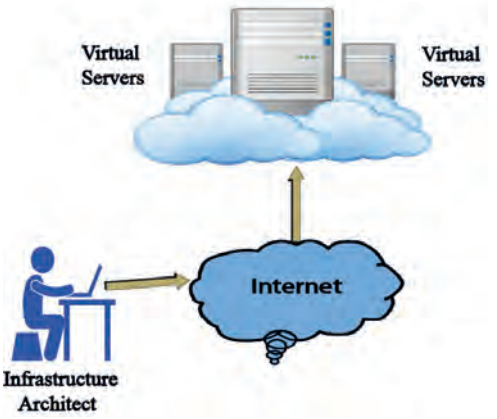


Figure 2.26. Infrastructure as a Service (IaaS)

Activity 2.3

1. Discuss the three types of cloud computing with their benefits
2. Why is SaaS becoming an increasingly popular delivery model for web services?
3. Mention any cloud-based services that are currently being used in Ethiopia through SaaS and PaaS cloud computing services models

2.3. Fog Computing

Brainstorming 2.3

- What are the difference and similarities between Cloud Computing and Fog Computing?

Fog computing is an extension of the cloud. Cloud Computing relies heavily on the bandwidth made available, which depends on the capacity of the network service provider. With billions of users processing, sending, and receiving data in and out of the cloud, the system becomes increasingly congested.

Fog computing uses the concept of 'fog nodes' which are located closer to the data source and have a higher processing and storage capability. Fog provides the

missing link for what data needs to be pushed to the cloud, and that can be analyzed locally, at the edge. This makes fog nodes to process data quicker than sending the request to the cloud for centralized processing.

What distinguishes fog computing from cloud computing is its closer proximity to small end-users, its wider consumer reach, and better mobility. Rather than requiring devices to go through the network backbone infrastructure, fog computing permits devices to connect directly with their destination with ease and allows them to handle their connections and tasks in any way they see fit. As a result, fog computing improves the quality of service, reduces latency, and enhance user experience.

Fog computing smoothly supports the emerging Internet of Things (IoT) physical things (vehicles, home appliances, and even clothes) that are embedded with sensors to enable them to send/receive data. This advantage makes it easier to run a real-time, Big-Data operation with the ability to support billions of nodes in highly dynamic and diverse environments.

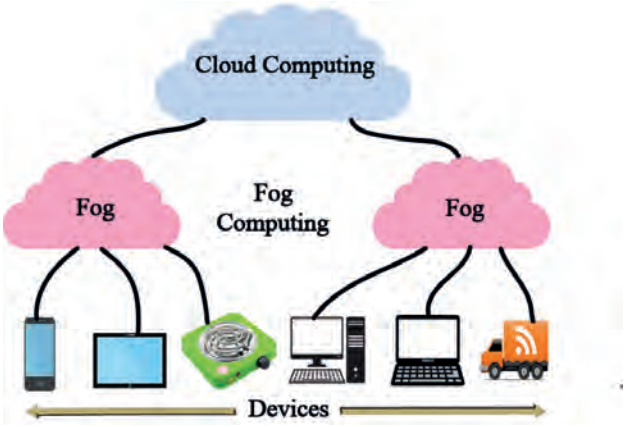


Figure 2.27. Fog Computing

For example – we can apply fog computing in video surveillance, where continuous streams of videos are large and cumbersome to transfer across networks.

Activity 2.4

1. Why do we need to use fog computing instead of cloud computing?
2. Due to the introduction of fog computing, what types of cloud services will be more efficient and usable? Bring real-life examples based on your previous online service user experiences.

2.4. Internet of Things (IoT)

Brainstorming 2.4

- Have you considered using the Internet for other purposes other than social media and information retrieval? How?

The Internet was created in the late 1990s and is now more than 30 years old, with over 2 billion people using computers, cellphones, and tablets to access it.

As per the definition of Oracle.com, ***Internet of Things (IoT)*** *is a network of physical objects or people referred to as "things" that are equipped with software, electronics, networks, and sensors to collect and exchange data* (Figure2.28). The term has recently been added to the Oxford dictionary and is defined as: *"The interconnection via the Internet of computing devices embedded in everyday objects, enabling them to send and receive data".*

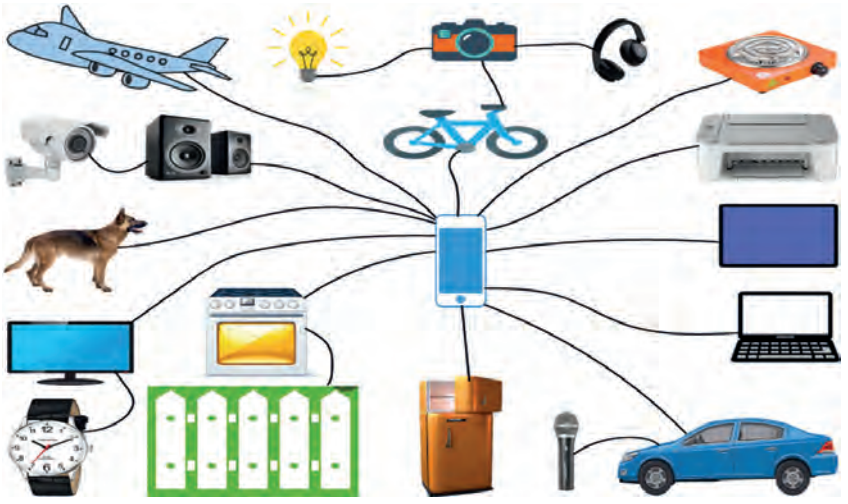


Figure 2.28. Internet of Things

IoT aims at extending Internet connectivity beyond computers and smartphones to other devices people use at home, or for business. The technology allows devices to be controlled across a network infrastructure remotely. As a result, it cuts down human effort and paves the way for accessing connected devices easily. With remote control, the devices can be used without involving human interaction. IoT makes things virtually smart through Artificial Intelligence algorithms, data collection,

and networks to enhance our lives.

Figure 2.29. IoT smart wearable and devices

IoT aims to provide Internet access to relatively simple equipment like a toaster from commonplace ones like PCs, mobile phones, and tablets. With the power of data collecting, AI algorithms, and networks, IoT transforms almost everything into a "smart" system. The thing in IoT can also be a person with a diabetes monitor implant, an animal with tracking devices, etc. Some examples of IoT are animal tracking devices, diabetes monitors, AC sensors to adjust the temperature based on the outside temperature, and smart wearables (See Figure 2.29).

Links

Refer to Grade 11 IT Textbook about Artificial Intelligence (AI)

When something connects with the Internet, it has both the capabilities of sending and/or receiving information and making it smart. IoT is creating lots of opportunities by linking computer systems with the real, and physical world.

For example : Just a few years ago, when you wanted to listen to your favorite songs on your mobile phone, you had to download each song. Nowadays, you can listen to any music you want without downloading it. All you need to do is access it where it is stored on the Internet.

2.4.1 Major advantages of IoT

Efficiency- We can gather trustworthy and secure real-time data thanks to IoT, and

we can then translate it into really useful information for enterprises. This helps to reduce the consumption of resources to improve our efficiency and productivity. For example, if you want to find the answer to a question, instead of browsing on your phone or turning your computer on, you can just ask a voice assistant such as Siri, Alexa, Google Home, Cortana, and more.

Technical optimization- You no longer have to operate multiple devices for each task manually if you use an IoT system because it allows you to control everything using a single device like your smartphone. You can easily control Television (TV) volumes, thermostats, and dim your lights, etc.

Convenience- IoT can provide quick and quality solution by improving customer experience.

Improved customer experience- As you can offer your targeted customers what they need, you can understand their pain points, and provide quick and quality solutions. This improves the customer experience.

Conservation- IoT helps to conserve the environment by monitoring traffic, water/ electricity usage, air quality, etc..Smart cities are developing and it also assists city planners and residents in finding solutions to problems.

Personalization- As IoT devices can collect your data, they can learn your preferences, likes, and dislikes quickly. As a result, they trigger more personalized services based on your choices.

Some Limitations of IoT

Security and privacy - IoT devices are evolving and becoming more widely used, making it difficult to keep the data they collect and transmit secure. IoT devices are being used in more sensitive industries, like healthcare and banking, which raises concerns about data protection. Global information privacy regulations are also becoming effective, so protecting data is not only beneficial from a commercial standpoint but is also required by law.

Connectivity and power dependence - For many devices to operate successfully, the internet and constant power are required. When either goes down, everything connected to it and the device also do.

Complexity and Integration - IoT is an advanced technology whose design is quite complicated. Plus, maintenance and deployment are also not easy for some IoT solutions. IoT protocols and standards are still subject to discussion, therefore devices made by various manufacturers may not be compatible with current hardware. It may be challenging to integrate effectively because each one may need various hardware connections and setups.

Higher Cost - IoT device deployment frequently requires a significant time and financial investment. There are numerous devices that need to be purchased, setup, and installed by professionals. This causes a subsequent exponential increase in cost.

2.4.2 How Does IoT Work?

A complete IoT system integrates four component.

1. **Sensors/devices-** First, sensors or devices collect data from their environment. Multiple sensors can be bundled together or sensors can be part of a device that does more than just sense things. For example, smart mobile phone is a device that may have multiple sensors (camera, accelerometer, etc.), but your phone is not just a sensor. There are plenty of sensor types which can be used based on the purpose.
    - Examples includes temperature sensors, proximity sensors, humidity sensors, smoke sensors, water quality sensors, infrared sensors, motion sensors etc.
2. **Connectivity-** The collected data is sent to the cloud through a variety of methods including cellular, satellite, Wi-Fi, Bluetooth, or connecting directly to the Internet via Ethernet. Choosing which connectivity option associated to the specific IoT application required.
3. **Data Processing-** The software processes the data after it is transferred to the cloud. This could be very simple, like making sure the temperature reading is within permissible limits, or highly complicated, like utilizing computer vision to detect things in the video (like intruders in your house). But what happens if the temperature gets too high or if someone breaks into your home? The users might steps in at this point as required.

4. **User Interface-** Next, the information is made available to the end-user through an alert to the user (email, text, notification, etc.). Depending on the IoT application, the user may also perform an action and affect the system. For example, the user might remotely adjust the temperature in the cloud storage via an app on their phone, or system could take actions automatically based on predefined rules. Example - Instead of calling or texting the user to alert of an intruder, the IoT system could be configured to automatically notify relevant authorities or agencies including security and legal offices.

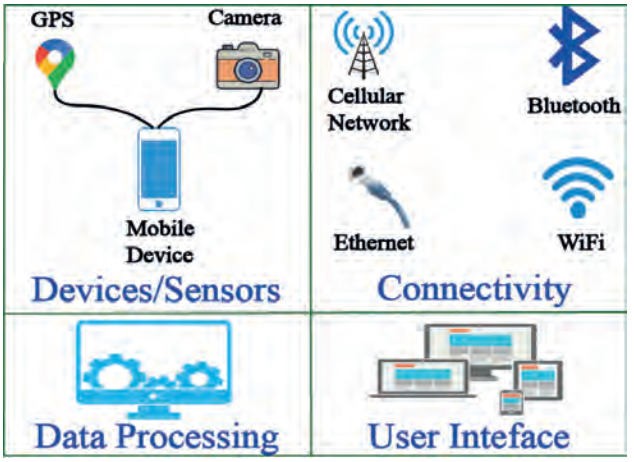


Figure 2.30. IoT components

For example, the user might remotely adjust the temperature in the cloud storage via an app on their phone, and some actions are performed automatically. Rather than waiting for you to adjust the temperature, the system could do it automatically via predefined rules. It instead of calling you to alert you of an intruder, the IoT system could also automatically notify relevant authorities or agencies as per the prior configuration.

2.4.3 Application of Internet of Things

Technology-based innovations like IoT are going to have a vital role in areas such as healthcare, economy, marketing, finance and banking, and government sectors. It uses advanced levels of automation, connectivity, and customization. See below for some of the common IoT application areas and their respective examples. Make further online research to understand the details of mentioned applications as required.

Smart Home- Smoke detector, light bulbs, appliances, door and window locks, pet monitoring camera, smart thermostat. (See Figure 2.31).

Figure 2.31. Smart Home

Smart City-Traffic Management, Waste management, and Water Distribution (Fig 2.32).

Figure 2.32. Smart City IoT Traffic Management

Supply chain- Real-time goods tracking and getting inventory information.

Connect Health- Patient care, calorie expenditure, heart rate pattern, temperature, activity levels, etc.

Connected cars- Automobile companies can automatically perform billing, insurance, parking, and more.

Smart outlets- turning a device on/off remotely, tracking the energy levels of a device, etc.

Generally, IoT affects many areas of day-to-day life and many social and technical issues can be solved by using the applications of IoT (See figure 2.33).

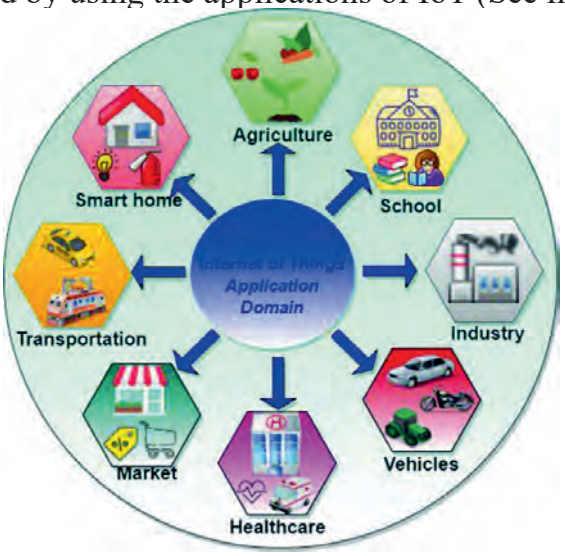


Figure 2.33 Common IoT Applications Areas

Optional Individual Case Study

- *You can think of any challenge in your community and plan to design an IoT system to solve it. Consult your IT teacher and other IT professionals for guidance and support on top of the available resources on the Internet. You can refer to Table 2.1 below for common IoT application areas and their uses.*

Table 2.1. Common IoT Application Areas and Use

Industry/Area	Use
Home	Control of heating, lights, door locks, etc.
Health	Remote patient monitoring etc.
Fitness and wellness	Tacking heart rate and training plans
Factory and industry	Production line control, asset tracking, etc.
Agriculture	Automatic watering, soil monitoring, etc.
Cars and roads	Connected cars
Smart cities	Traffic management, parking space, tracking, and availability.

Activity 2.5

1. Explain the potential of IoT for smart home applications. Discuses types of routine activities at home that can supported and managed be replaced by smart home IoT applications.
2. Form a group of three students. Select any industry and present the potential applications of IoT with practical scenarios.

Unit Summary

In this unit, you have learnt about:

- Bigdata and its relation to data science
- common Bigdata characteristics
- advantages of Bigdata
- application areas of Bigdata in different sectors
- cloud computing and its benefits
- cloud computing types, services, and deployment models
- cloud-based applications examples
- basics of Fog computing and its relation to cloud computing
- the Internet of Things (IoT)
- major advantages and applications of IoT

Key Terms

Big Data - refers to complex and large data sets that have to be processed and analyzed to uncover valuable information that can benefit businesses and organizations

Cloud Computing - The distribution of various services, such as data storage, servers, databases, networking, and software, through the Internet

Fog Computing - is a decentralized computing infrastructure in which data, compute, storage and applications are located somewhere between the data source and the cloud.

Infrastructure as a Service- is a form of cloud computing that provides virtualized computing resources over the internet.

Internet - a global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols.

Internet of Things - describes physical objects with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks.

Platform as a Service - is a type of cloud computing that offers users a whole cloud platform, including hardware, software, and infrastructure, for creating, administering, and using applications.

Software as a Service - is a way of delivering applications over the Internet—as a service where users can avoid complicated software and device maintenance by just accessing software via the Internet rather than installing and maintaining it.

Review Questions

Part I: Choose the correct answer from the given alternatives.

1. Which one of the following is not correct about basic views of Bigdata?
   A. Bigdata refers to a massive amount of data that keeps on growing exponentially with time.
   B. Bigdata is processed or analyzed using conventional data processing techniques.
   C. Bigdata includes data mining, data storage, data analysis, data sharing, and data visualization
   D. Bigdata is an all-comprehensive one including data and data frameworks.

2. Which one of the following characteristics of Bigdata refers to the speed at which data is being created in real-time?
   A. Velocity
   B. Volume
   C. Variety
   D. Value

3. As IoT exposes a huge amount of detailed personal information, it could create many privacy issues.
   A. True
   B. False

4. One of the following is NOT among the four distinct components of IoT to have an integrated and complete IoT system.
   A. User Interface
   B. Sensors / Devices
   C. Connectivity
   D. Word Processing Applications

Part II (a): Match items in Column A to corresponding answers in Column B

A	B
1. Hybrid Cloud	A. SaaS
2. PaaS	B. combination of Public Cloud and Private Cloud
3. Microsoft Office 365	C. Windows Azure
4. Public Cloud	D. Dropbox
	E. Service providers' resources are shared across clients.
	F. Robotics

Part II (b): Match the following industries under column A to their respective IoT applications under column B

A	B
1. Home	A. Traffic management
2. Health medical	B. Smoke Detector
3. Fitness and wellness	C. Tacking heart rate
4. Factory and Industry	D. Soil Monitoring
5. Agriculture	E. Remote Patient Monitoring
6. Smart Cities	F. Asset Tracking
	G. Driverless car
	H. Drones

Part III: Give short answers to the following questions

1. What is the aim of IoT? What makes it different from networking?
2. Mention two practical examples of IoT applications.
3. Discuss applications of Bigdata in the education sector with examples.
4. Mention common advantages of fog computing services.

Part IV: Case Study

1. Form a group and find out a local organization or community groups that could benefit from different IoT applications. Explore the organization and community groups major activities and address the following questions.
   a. why do you think IoT can help their operations?
   b. what types of technologies / infrastructures required to implement the selected IoT applications?
   c. what benefits can be obtained by the implementations of IoT?
   d. what limitations are expected during the implementation of the selected IoT implementation?

   Prepare a brief report document and present your groups work to the class following your teacher/s direction and schedule. It is advisable to contact your course teacher and/or industry experts in your community for any support if required.

UNIT 3

DATABASE MANAGEMENT SYSTEM

Learning Outcomes

At the end of this unit, students will be able to:

- Describe relational database management system.
- Explain fields and records.
- Apply SQL data definition language(DDL) to create tables in a database.
- Differentiate between SQL data definition language(DDL), data manipulation lanaguage (DML), and data query language(DQL).
- Apply SQL DML to manipulate records in tables.
- Apply SQL DQL to query records of tables.

Unit Overview

In today's world, almost all information is stored in databases. In Grade 11, you learned that databases are an important part of organizations to manage their data. You also learned how to design a database. A database has a data model that determines the manner in which data can be stored, organized, and manipulated in the database. The focus of this unit is on the creation of a database. Technologies, knowledge, and skills needed to design the database are also discussed. The unit also say how SQL (Structured Query Language) allows you to manipulate records of a database.

3.1. Overview of Relational Database Management System

Brainstorming 3.1

- What is a relational database management system?

The term Relational Database Management System (RDBMS) usually refers to various types of software systems developed in order to manage databases. RDBMS is used to create, maintain, and provide controlled access to a relational database.

A relational database is based on a relational data model. Data are stored in a two-dimensional table, which contains columns or fields and rows or records. Each column of a table represents an attribute or data value, and each row in a table represents a tuple or record.

A database contains one or more tables that maintain records. Attributes are the set of properties to describe the instances of the entity. For example, a student can be an entity. The attributes of a student can be described in terms of student id, name, age, grade level and sex. The table shown in Figure 3.1 below has five attributes (or pieces of data). A record is a row or a tuple in the table. It contains a single data value in each column.

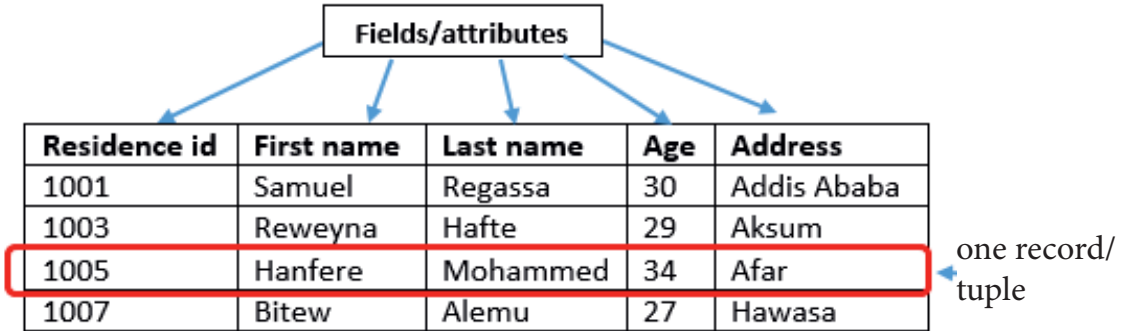


Figure 3. 1 Example table with fields and records

Each field in a table has to be given a name and data type. A data type is the type of data value you want to store in the field. For example, to populate this database you would use residence id to populate the residence id field. Here, we know that a residence id is mentioned as a four-digit number. The second field, first name, is used to store the first name. Name is an attribute stated in text form.

In the database, the data type best describes the fields in the table. For example, referring to Table 3.2, age can be input by an integer data type, price by float or real number, etc. (See Table 3.1).

Table 3. 1 Sample fields with appropriate data types and data values

Fields	Data type	Data values (Example)
Age	Integer	19
Name	Text	Abdu
Price	Real or currency	32.5
Date of Birth	Date	10/12/2008
Height	Integer or real	175 or 165.6
Grade level	Integer	11 or 12
Telephone number	Text	+251 900 00 00 00

All Database Management System(DBMS) provide data types from which to select and use to define the data type for fields of a table. For example, in MS Access, when you create a new database table, you specify what type of data is to be stored in each field. The data types include text, integer, real numbers, dates, currency and yes or no fields (See Figure 3.2).

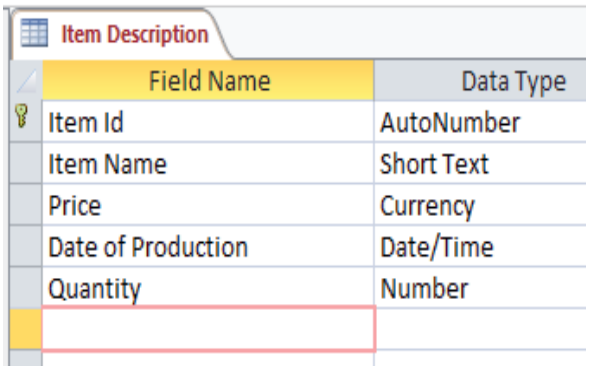


Figure 3.2 Sample data type in MS Access

Links

See Table 3.3 in section 3.2.2 of this unit for detailed descriptions of data types in MS Access.

Activity 3.1

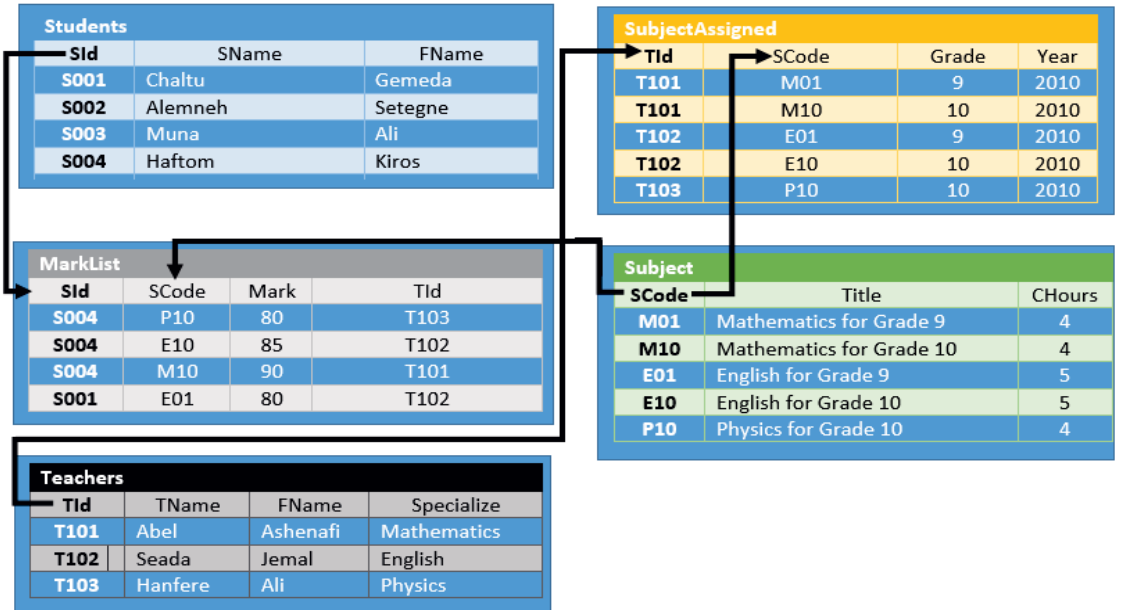
1. Give sample data value based on the data type described in the following table.

Fields	Data type	Example
Book Title	Text	
Email	Text	
Sales Date	Date	
Body temperature	Integer or real	
Weight	Integer or real	
Cost of material	Real or currency	
Expire date	Date	
Opening hour	Time	

2. Given a patient table,write three fields and their data types to describe a patient.

The database in Figure 3.3 below has five tables: Students, MarkList, Subject, SubjectAssigned and Teachers. For example, the **'Students'** table contains three fields: Sid, SName and FName. These are attributes that describe the student entity. The **Students** table has four records of which the first record has data value of S001, Chaltu, and Gemeda. In the **Students** table, the field Sid is a primary key. The data value of Sid is unique, meaning that a record with Sid S001 cannot be duplicated. In relational database, each table that has a key field (primary key) uniquely identifies each record. This key field can be used to create relationship and link one table of data to another. The **Students** table is linked with MarkList through the Sid field. This table, MarkList, consists of students' marks for the enrolled courses. A student who took four courses could have four mark records with same Sid (e.g. S004 in MarkList). Therefore, Sid in the table MarkList is not a unique key or primary key. The Sid in MarkList is referred to as a foreign-key. This key is used to create a relationship with the **Students** table which contains Sid as primary key.

Relational databases are the most popular databases have been widely used for more than four decades. Examples of popular RDBMs include Microsoft Access, Oracle, Microsoft SQL Server, MySQL, SQLite, IBM DB2, and PostgreSQL.



Students

SId	SName	FName
S001	Chaltu	Gemeda
S002	Alemneh	Setegne
S003	Muna	Ali
S004	Haftom	Kiros

SubjectAssigned

TId	SCode	Grade	Year
T101	M01	9	2010
T101	M10	10	2010
T102	E01	9	2010
T102	E10	10	2010
T103	P10	10	2010

MarkList

SId	SCode	Mark	TId
S004	P10	80	T103
S004	E10	85	T102
S004	M10	90	T101
S001	E01	80	T102

Subject

SCode	Title	CHours
M01	Mathematics for Grade 9	4
M10	Mathematics for Grade 10	4
E01	English for Grade 9	5
E10	English for Grade 10	5
P10	Physics for Grade 10	4

Teachers

TId	TName	FName	Specialize
T101	Abel	Ashenafi	Mathematics
T102	Seada	Jemal	English
T103	Hanfere	Ali	Physics

Figure 3. 3 Relational Database model with sample data.

Activity 3.2

1. Describe the main features of RDBMs.
2. What is field in a table? Or give an example.
3. What is record in a table? Give an example.

3.2. Database Manipulation Using SQL

Brainstorming 3.2

- What are the similarities and differences between SQL and other programming languages?

SQL (**S**tructured **Q**uery **L**anguage) is a standard language for accessing and manipulating a database. SQL is a special-purpose query language meant for interacting with relational databases such as Microsoft Access. Understanding how SQL works can help create better queries and make it easier to understand how to fix a query that is returning unwanted results.

SQL consists of a number of commands with further options to allow you to carry out your operations with a database. Based on their purposes, three categories of SQL commands are presented below.

- **Data Definition Language (DDL)**: DDL contains commands that allow you to create or modify tables and establish relationship between tables in your database structure.
- **Data Manipulation Language (DML)**: DML contains commands that are used to manage the database by performing operations such as inserting, updating, deleting, and navigating through data.
- **Data Query Language (DQL)**: is used for querying or selecting all or subsets of data from a database.

In addition to the above SQL commands, the DBMS may give you other sets of commands to provide extra features. However, the three categories of commands are common and applicable in all DBMSs.

Activity 3.3

1. What SQL command can be used to:
   - create a table?
   - modify a table?
   - delete a table?
   - add a record to a table?
   - modify a record in a table?
   - remove a record in a table?
2. Categorize commands in question 1 under DDL, DML and DQL.

3.2.1 Using SQL in Microsoft Access

Brainstorming 3.3

- What applications are included in Microsoft Office package?

Microsoft Access is the default relational database program installed with the Microsoft Office package. It offers the functionality of a database, and the

programming capabilities to create databases easily and navigate records. SQL command in Microsoft Access can be used to implement and manipulate.

Notes

- *Check Microsoft Access is installed and its version. From Windows start button, click 'Windows' icon --> then 'All apps' icon --> from the list find and click on 'Microsoft Office'. You will see 'Access' followed by year (version) in the list if it is installed.*

Create a database

Launch Access from the Start menu just by typing access in the Search box as indicated in Figure 3.4. After starting Access, you are asked to write a name for your database.

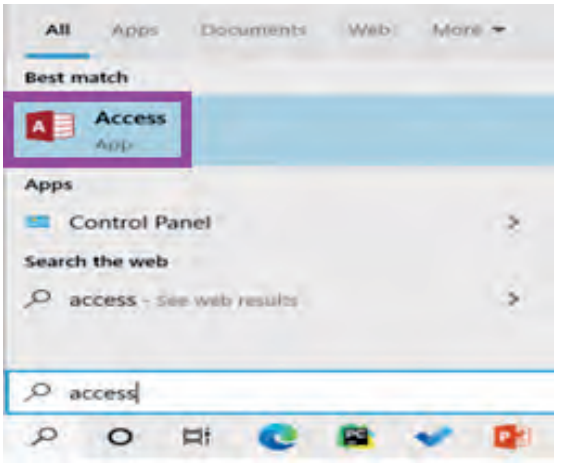


Figure 3.4 Launch Access from start menu

There are two ways of creating a database in Microsoft Access:

1. Create a Database from template.
2. Create a blank database.

Select the second option to create a database from the scratch.

How to Create Table

The first step, after creating your database in Microsoft Access, is creating a Table for your data. Here are three ways of creating database tables in Access.

1. Create a Table from **Design View**
2. Create a Table from **Datasheet View**

3. Create table from **SQL**

Figure 3.5 Start a Query in MS Access

From the three ways of creating database tables, the third is discussed in this section.

How to write a query

In order to start writing a query, first you click on the **Create** tab, then click on the **Query Design** button (See Figure 3.5).

Notes

- Close the Show Table dialog box that appears when you click the Query Design in the previous step.

Next, click on the **Dropdown** of the **SQL View** that appears on the left corner on the **Window**s and from the options displayed, select the **SQL View** (as shown in Figure 3.6). This opens the query writing windows. On this window, you can write your SQL commands (CREATE, SELECT, UPDATE, INSERT etc).

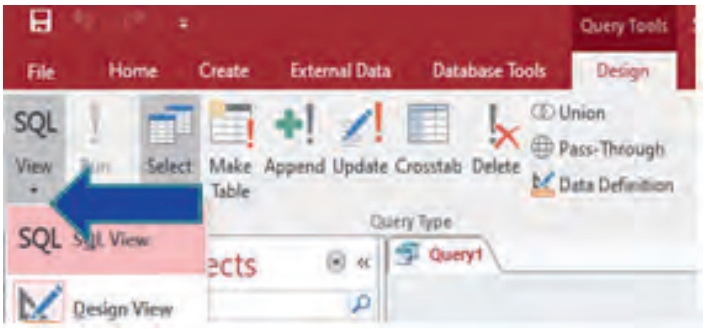


Figure 3.6 SQL View in MS Access

How to Run SQL Commands

After writing your query using SQL commands, you can save or run your query to

see the results of the query. If there are errors in your query, Access shows you a pop-up windows based on the error type. There are alternatives to run your SQL commands.

Alternative 1: On the **Design** tab, in the **Results** group, click on the **Run** option (See Figure 3.7).

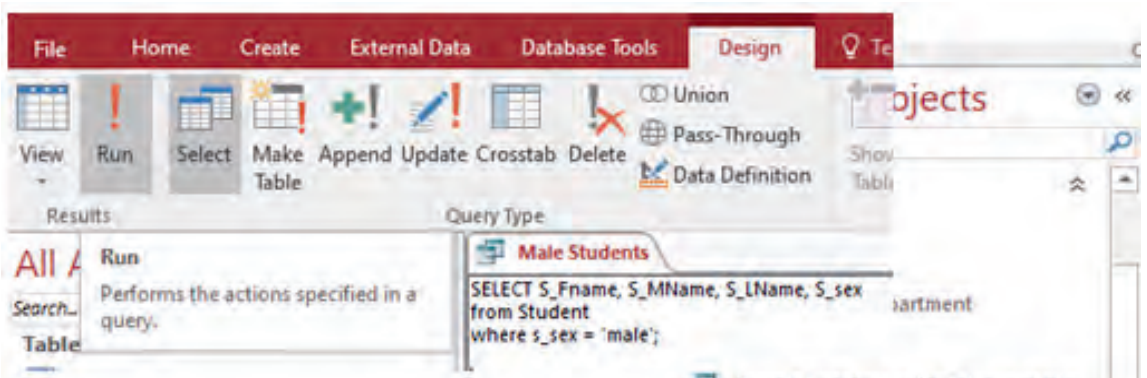


Figure 3.7 Run view in MS Access Design tab

Figure 3. 8 Queries view in MS Access (Double Click the Query highlighted

Alternative 2: Double-click the query you want to run. This works if you have already saved your query. For example, clicking on **Male Students** in Figure 3.8) run the query that retrieves list of male students from the database.

Activity 3.4

- Perform the following using Access:
  - Open MS Access.
  - Create a blank database, name it 'School'.
  - Create a new query and save as 'student query'.

3.2.2 Data Definition Language (DDL)

DDL is part of SQL that is used to create and restructure a database. Some of the most fundamental DDL commands include CREATE TABLE, ALTER TABLE, and DROP TABLE

CREATE TABLE Command

The CREATE TABLE command is used to create a new table in a database. The CREATE TABLE command provides various options to create a table. Table 3.2 illustrates the general syntax that is used to create a table.

Table 3. 2 SQL syntax for CREATE TABLE command

```
CREATE TABLE table_name(
Column1 datatype [Primary Key],
[Column2 datatype][REFERENCES table_name2(Column1)],
[Column3 datatype], [Column4 datatype]…)
```

Where:

1. CREATE TABLE is SQL command to create a table.
2. table_name is a name for the table. E.g. Farmer
3. Column1, Column2, Column3 are fields of the table. E.g. name, age, subject, etc.
4. Optional [Primary Key] is a command to set unique identifier.
5. datatype describes the type of data a field can store. E.g. CHAR, INT, etc.

Notes

- Each column in a table is required to define a name and a data type. Various DBMS might use different data type names. In some case, the name could be the same but different in their size and other details. Therefore, you need to always check their documentation.

Some of the more specialized data types that are used in SQL statement in Access are listed in Table 3.3.

Table 3. 3 Common data types used in MS Access

MS Access SQL data types	**Description**	**Allowable data value or size**
CHAR	This is used for text or combinations of text and numbers, including numbers which do not require calculating (e.g. phone numbers). If field size is ignored (e.g. Id CHAR), the field Id stores up to 255 characters. Alternatively, you can limit the size like Id CHAR(20). In this case the Id can store a maximum of 20 characters.	Up to 255 characters.
TEXT	This is used for lengthy text or combinations of text and numbers. E.g. name TEXT or name TEXT(20). The later data type limit the size of characters that can be stored to be 20.	Up to 63, 999 characters.
INTEGER or INT	This data type allows to store long integer data. E.g. populationSize INTEGER or populationSize INT. This data type does not allow users to specify or limit size.	-2,147,483,648 to 2,147,483,647
DATETIME	This is used to store Date and time values for the years 100 through 9999. E.g. DoB DATETIME. The date can be described in 'dd-mm-yyyy' or 'yyyy-mm-dd' form. Both date and month should be described with two digits, whereas year should be described with four digits. E.g. '10-12-2020' or '2020-12-10' where 10 describes date, 12 describes month and 2020 describes year. The Time value can be described in 'hh:mm' or 'hh:mm:ss' form. E.g. '10:20' or '10:20:10'.	DATE from 01-01-100 to 31-12-9999 and TIME from 01:00:00 to 23:59:59

MS Access SQL data types	Description	Allowable Data Value or Size
CURRENCY	This is used for currency or monetary value. Currency value and numeric data are used in mathematical calculations.	8 bytes
COUNTER	This is a unique sequential (incremented by 1) number or random number that is assigned by Microsoft Access whenever a new record is added to a table.	4 bytes
Yes/No	Yes and No values and fields which consist of only one of two values (Yes/No, True/False, or On/Off).	1 bit.

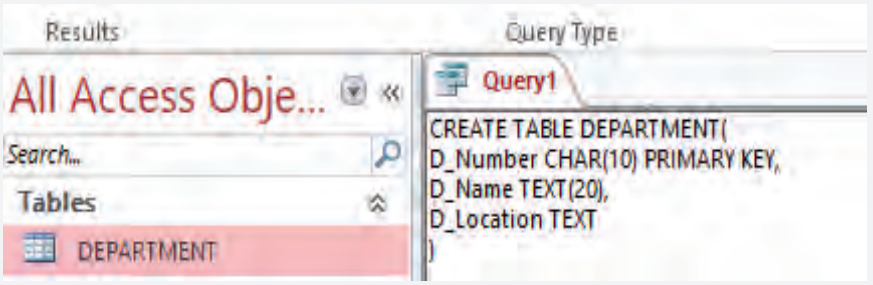
Create tables DEPARTMENT and COURSE (See Figure 3.9). The DEPARTMENT table is created with three attributes namely, D_Name, D_Number and D_Location, where D_Number is the primary key. The COURSE table has five attributes: Contact_Hr, C_Name, C_Description, C_Department, and C_Code. The primary key for the COURSE table is C_Code.

Do not forget to make the data type of the C_DEPARTMENT to be similar with D_Number as these attributes are used for creating relationship between the two tables (i.e. C_DEPARTMENT TEXT(10) REFERENCES DEPARTEMENT(D_Number). They refer to the same thing but named differently in the two tables.

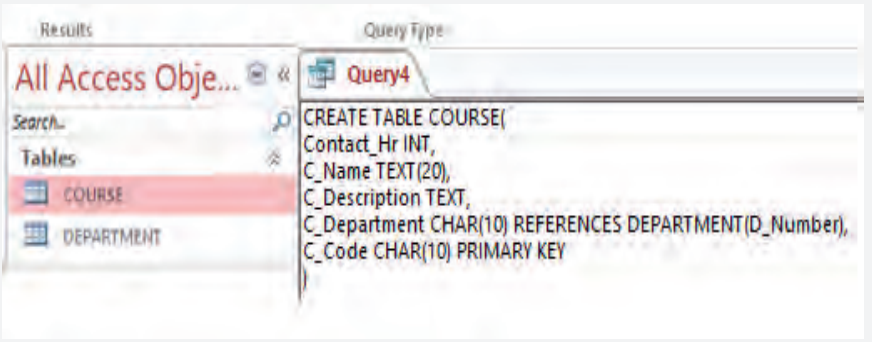
Notes

Figure 3.9 demonstrates CREATE TABLE COURSE with primary key (C_Code) and FORIGEN KEY (C_Department). The C_Department in the COURSE table and the D_Number in the DEPARTMENT table have the same data type. To create a relationship, the C_Department should be followed by the keyword REFERENCES. Then write the DEPARTMENT, which defines the D_Number as PRIMARY KEY. Enclose the D_Number with parenthesis - C_Department TEXT(10) REFERENCES DEPARTEMENT(D_Number).

If you create a relationship, as seen in earlier example, do not add data to the table until a relationship is set up. Otherwise, it generates error if the requirement is not met.



The figure shows SQL command to create DEPARTMENT table with three attributes and the table generated.



The figure shows SQL command to create the COURSE table with five attributes and the table generated.

Figure 3. 9 A Query view in MS Access to write SQLs Statements to create DEPARTMENT (upper) and COURSE (lower) tables

Notes

While using the CREATE TABLE command, the table name should be followed by opening parenthesis, and at the end there should be a closing parenthesis. The definition of each field should be separated by a comma. Each column should specify appropriate data type. When you specify character size (e.g. C_Name TEXT (20)) make sure that the data value of the field cannot exceed the size specified. Otherwise, the data value after the size specified is ignored. For example, if your data value is '*Introduction to Civics and Ethical Education*' only the first 20 characters will be stored in C_Name (i.e. '*Introduction to Civi*'). Space is counted as one character.

Activity 3.5

1. Create the following three tables using SQL commands Access:

   - Student table with attribute – student id (CHAR(10)), name (CHAR(20)), sex(CHAR(1)), age(INTEGER), grade level (INTEGER). Set the Student id as primary key.

   - Teacher table with attribute – teacher id (CHAR(10)), name (CHAR(20)), sex(CHAR(1)), age(INTEGER), specialization (CHAR(15)). Set the Teacher id as primary key.

   - Grade table with attribute – student id (CHAR(10)), teacher id (CHAR(10), course code(CHAR(20)), mark(INTEGER). Use course code as a FOREIGN KEY to create a relationship with the COURSE table created in Figure 3.9.

2. Run the SQL command and see the resulting tables created (student, teacher, and grade).

3. The following CREATE statement has a problem (i.e. has syntax error). Trace the problem and rewrite the statement to fix the error

   - CREATE TABLE Patient (pid INTEGER, P_Name TEXT P_Age INT

   - CREATE Physician (id CHAR, name TEXT, DoB DATETIME, PRIMARY KEY(id))

   - CREATE TABLE Treatment (ptid INTEGER REFERENCES Patient pid, drId CHAR REFERENCE id)

4. Write the CREATE table SQL statement with appropriate data type and size based on the records in the following Physician table.

   Hint: whenever you specify data size, use the value with the maximum character size.

employee id	Name	position	bank account	bank
1	Herpesa Mekonnen	Internist	7555666111	Wegagen
2	Yohanna Abrha	Pharmacist	1022405505500	CBE
3	Abdela Kibomo	Nero Surgeon	23567434566	Awash
4	Selamawit Kassu	Head of Clinic	4509008755	Dashen
5	Jemila Zeynu	Surgical Physician	66688979045	COOP
6	Animaw Mekuwanint	Paediatrician	1000010150198	CBE
7	Sirage Aliyu	Psychiatrist	1080880880880	Abyssinia

ALTER TABLE command

Once a table is created, it can be modified using the **ALTER TABLE** command. Using the **ALTER** command, you can add column(s), drop column(s), and change column definitions. It is also used to establish relationship between tables.

If in Figure 3.9, the datatype for D_Number is CHAR(10) and C_Department is CHAR(15), this generates an error while you create relationship between the two tables. Thus, modify the field C_Department in the COURSE table with CHAR(10). To do this, write:

ALTER TABLE COURSE ALTER COLUMN C_Department CHAR(10)

Or drop the field C_Department and

ALTER TABLE COURSE **DROP COLUMN** C_Department**;**

Then, add C_Department again like this:

ALTER TABLE COURSE **ADD COLUMN** C_Department **CHAR(10);**

If you have missed setting a primary key while creating a table, the ALTER command can be used to modify the column as primary key. Assume you have not set a primary key for the TEACHER table. Modify the teacher_id and set it as primary key as follows:

ALTER TABLE TEACHER ADD PRIMARY KEY (teacher_id)

ALTER command to create relationship

The following ALTER command modifies the COURSE table to create a

relationship with DEPARTMENT table through D_Number which is the primary key in DEPARTMENT table.

ALTER TABLE COURSE **ADD FOREIGN KEY (**C_Department**)**
REFERENCES DEPARTMENT **(**D_Number**);**

Activity 3.6

1. Modify the Physician table in question 4 of Activity 3.5 based on the following descriptions using ALTER command:
   - Modify 'employee id' datatype to CHAR of size 7. E.g. 'Py/1001'.
   - Modify 'employee id' as PRIMARY KEY.
   - Add a new column for date of hire of the physician with DATE-TIME datatype.
2. Modify the Student and Teacher tables in question 1 to create the following relationship using the ALTER command:
   - Add student_id for the Student table as foreign key of table grade, and
   - Add teacher_ id for the Teacher table as foreign key of table grade.

DROP TABLE command

Dropping a table is actually one of the easiest thing to do. Sometimes you might create a table that you do not want, or you want to remove after creation. Use the DROP TABLE command to delete the table already created. For example, if you want to delete COURSE table, use the following SQL command.

DROP TABLE COURSE

Caution - Remember that if you delete a table with record, you cannot get it back.

Activity 3.7

- Create a table with two columns and write an SQL statement to drop the table.

Create Relationship Visually using 'DATABASE TOOL' tab in MS Access

Previously you saw how to use SQL ALTER command to create a relationship between tables. Alternatively, you can create a relationship using graphical database tool in MS Access. A relationship in MS Access helps you combine data from two or more different tables. Each relationship consists of fields in two tables with corresponding data. The relationship can be one-to-one, one-to-many, or many-to-many.

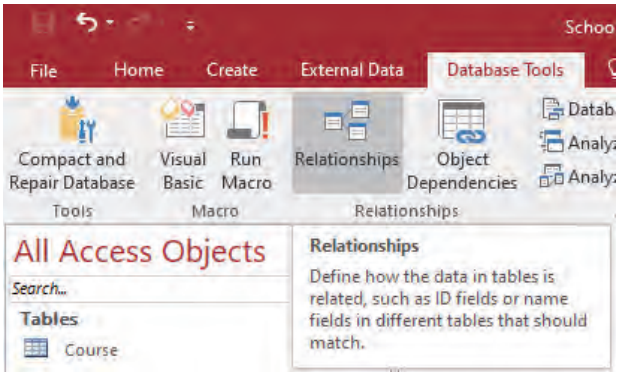


Figure 3. 10 Start Relationships in MS Access

Links

Refer to Unit Three of Grade 11 IT Textbook about types of relationship in relational database model for further reading.

Steps to create Relationship

- On the 'Database Tools' tab, in the 'Relationships' group, click 'Relationships'.
- On the pop-up 'Show Table', select one or more tables or queries and then click '**Add'**. After you have finished adding tables and queries to the Relationships document tab, click '**Close'** (See Figure 3.10).

Drag a field (typically the primary key) from one table to the common field (the foreign key) in the other table. To drag multiple fields, press the CTRL key, click each field, and then drag the Primary key over the foreign key.

The diagram below (Figure 3.11) shows the relationships among the tables of the school database. The Student, Course, and Teacher tables are related to the DEPARTMENT table using the D_NUMBER. D_NUMBER is a primary key in the DEPARTMENT table, whereas it is a foreign key on the remaining three

tables to create relationships. Similarly, T_IDNO is a foreign key in Offer_Course table, and C_Code is a foreign key in both Offer_Course and Take_Course tables. S_IDNO is a foreign key in Take_Course table. Note that the primary keys (e.g. D_Number in the DEPARTMENT table) and the foreign keys (e.g. T_department in the Teacher table) have the same data type. Also, data should not be added until the relationships are set up because if, for example, you add records and then setup relationships where the records do not meet the requirements, you get errors.

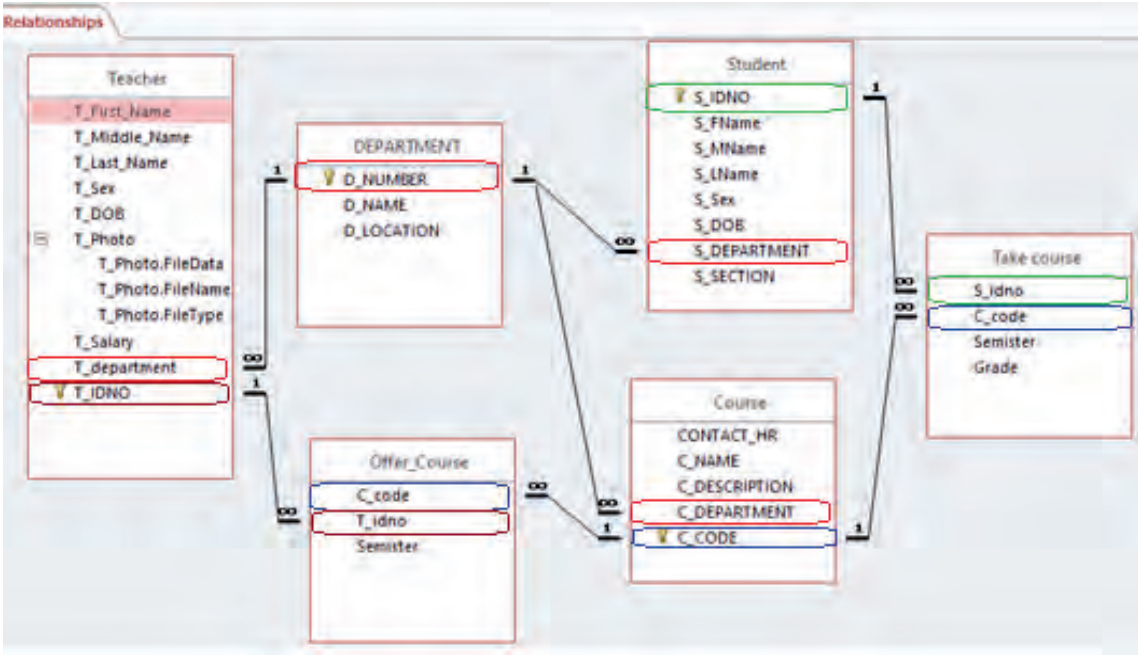


Figure 3. 11 Relationship in relational database mode

Notes

The infinity symbol (∞) and 1 in Figure 3.11 describe the relationship between the tables indicated. ∞ is used to describe the many side whereas the one side is described by '1'. For example, the relationship between Department and Teacher is one-to-many. That means, one record in the DEPARTMENT table is linked to many records in the Teacher table. Thus, the many side is indicated with infinity symbol and one side is indicated by '1'.

An example of a relational database model is shown in Figure 3.12. Note that all related tables have a common data item (key field) that enables information stored in one table to be linked with information stored in another.

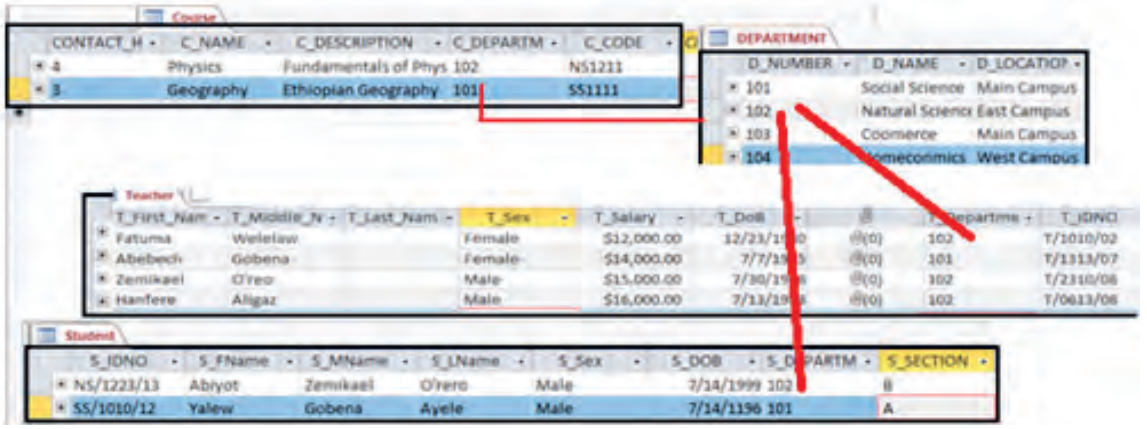


Figure 3. 12 Records in Relational Database model.

Activity 3.8

- Open the database you have created in question 1 of Activity 3.5.
  - Click on 'DATABASE TOOLS' tab, then Relationships. Do you see any relationship diagram? Discuss the relationships that has been created?
  - Delete the relationship by selecting the line that connect the Student table with grade table, and grade table with the teacher table. Then create the relationship again by clicking 'Relationships'.

3.2.3 Data Manipulation Language

Brainstorming 3.4

- What is data manipulation?

Data Manipulation Language (DML) consists of commands that allow you to manage the database by performing operations such as inserting, updating, deleting, and navigating through data. The DML commands include **INSERT, UPDATE and DELETE.**

INSERT command

INSERT command helps to insert new records to a table. The INSERT command can be used in one of the two options shown in the table below. Option 1 is used when the values of all fields are to be replaced by new values, whereas Option 2 is used when only selected fields are to be replaced.

Notes

Did you know that while using the INSERT INTO command, the column value should match the column data type of the table (e.g., you cannot insert a text value into a numerical field). All string values to be inserted in a table should be quoted with single or double quotation marks (e.g. 'Chemistry' or "Chemistry"). But if the field is defined as numeric, like Age INT, the value to be inserted does not need quotation marks(e.g. 20).

Option 1:

INSERT INTO table_name **VALUES** (value1, value2, value3 ...)

Option 2:

INSERT INTO table_name (column1, column2…) **VALUES** (value1, value2 …)

Example 1: Insert into a COURSE table

INSERT INTO COURSE **VALUES** (3, 'Chemistry', 'Organic chemistry', '105', 'NS1421');

The above example inserts a record into a COURSE table. The COURSE table (See Figure 3.12) has five attributes: credit hours, course name, course description, department, and course code. Accordingly, each value listed in the INSERT command is inserted into the corresponding column in the COURSE table. Except for the column credit hours (INT datatype), all fields in the COURSE table are string data type. Therefore, the data value '105' is considered a string rather than a numeric value.

Example 2: Insert into a DEPARTMENT table

INSERT INTO DEPARTMENT (D_Number, D_Name, D_Location)
VALUES ('D101', 'Biology', 'NBR First Floor')

Example 2 demonstrates inserting a record into the DEPARTMENT table using option 2 INSERT command syntax. Thus, next to the table name – DEPARTMENT, columns' names are listed being separated by commas and enclosed by parenthesis. The VALUES part is similar to the demonstration given in example 1. All fields that are not mentioned on the INSERT statement are left blank, or default value is recorded. For example:

```
INSERT INTO  COURSE (Contact_Hr, C_Name, C_Department, C_Code)
VALUES (4, 'Biology', '102', 'NS2323 ');
```

In the above INSERT statement, the value of C_DESCRIPTION is not mentioned on the query. As a result of this, while you run the query, the value of the course description for Biology is left blank in the table (See Figure 3.13)

CONTACT_HR	C_NAME	C_DESCRIPTION	C_DEPARTM	C_CODE
2	Accounting Pri	Accounting principle c	104	COM1212
3	Financial Accou	accounting finance	104	COM1312
4	Physics	Fundamentals of Phys	102	NS1211
4	Biology		102	NS2323
3	Geography	Ethiopian Geography	101	SS1111

Figure 3. 13 Course Table after Inserting Records

Activity 3.9

1. Insert the following records in the Student, Teacher and Grade tables you have created in Activity 3.5

Student_id	Name	Sex	Age	Grade_level
1001/2013	Halima Ali	F	16	11
1002/2013	Demess Lema	M	16	11
1010/2013	Shewit Belay	F	17	12
1011/2013	Kena Elias	M	17	12

- *Insert into teacher table*

Teacher_id	Name	Sex	Age	Specialization
2001/2010	Roman Nesibu	F	31	Physics
2002/2010	Helen Alemu	F	30	ICT
2010/2010	Belay Getaneh	M	30	Geography
2011/2010	Demeke Mulat	M	31	Mathematics

- *Insert into Grade table*

Student_id	Teacher_id	Subject	Mark
1001/2013	2001/2010	Physics	80
1002/2013	2002/2010	ICT	85
1010/2013	2010/2010	Geography	79
1011/2013	2011/2010	Mathematics	90

2. What is the problem with the following INSERT statement? Rewrite the INSERT statement to fix the error.

   - *INSERT IN TO GRADE VALUES('1012/13', '2023/2010', 'English')*
   - *INSERT INTO TEACHER VALUES('teacher_id', 'Sex', 'Age')*
   - *INSERT INTO STUDENT (Student_id, Name, Sex) VALUES('1015/13', 'Yared')*

3. Write INSERT statements to insert the following records.

   - *Student id, name and age for three students (1001/2014, Meymuna, 20), (1002/2014, Alehegne, 20), and (1003/2013, Kello,19).*

UPDATE command

The UPDATE command does not add new records to a table, nor does it remove records. It simply updates existing records in the table. The UPDATE command is used to change a value of one or more fields in an existing table row or number of rows. The general syntax of an UPDATE command is given below.

```
UPDATE table_name
SET column1 = value1, column2 = value2...
WHERE condition;
```

Notes

- Before you attempt to update any records you need to know that you are modifying the content of one or more records of a table.

Teacher

T_First_Nam	T_Middle_N	T_Last_Nan	T_Sex	T_DOB		T_Salary	T_de	T_IDNO
Haleluya	Biniam	Kiflu	Male	6/13/1988	(1)	$12,000.00	102	Teach/0613/08
Fatuma	Welelaw	Abdi	Female	12/23/1980	(0)	$10,000.00	101	Teach/1010/02
Abebech	Gobena	Ayele	Female	7/7/1985	(0)	$14,000.00	101	Teach/1313/07
Zemikael	O'rero	Samuel	Male	7/30/1978	(0)	$8,000.00	102	Teach/2312/91

Figure 3. 14 Teacher's Table before Update

Example:

If you execute the following UPDATE statement, then teachers with salary less than 10,000 are updated to 15,000 in T_Salary column. See Zemikael's salary in Figure 3.14 and Figure 3.15.

UPDATE Teacher **SET** T_Salary = 15000 **WHERE** T_Salary<10000;

Teacher

T_First_Nam	T_Middle_N	T_Last_Nan	T_Sex	T_DOB		T_Salary	T_de	T_IDNO
Haleluya	Biniam	Kiflu	Male	6/13/1988	(1)	$12,000.00	102	Teach/0613/08
Fatuma	Welelaw	Abdi	Female	12/23/1980	(0)	$10,000.00	101	Teach/1010/02
Abebech	Gobena	Ayele	Female	7/7/1985	(0)	$14,000.00	101	Teach/1313/07
Zemikael	O'rero	Samuel	Male	7/30/1978	(0)	$15,000.00	102	Teach/2312/91

Figure 3. 15 Teacher's Table after Update

Activity 3.10

- UPDATE the following records in the Student and Teacher tables (See Activity 3.9)
  - Modify student name, Halima Mohammed where student_id is 1001/2013.
  - Modify specialization, Economics where teacher_id is 2010/2010

The UPDATE command can also be used to update multiple rows at the same time. This is done by selecting many rows using the WHERE clause. It is the WHERE clause that determines how many records are updated. For instance, the following query updates all female salary to $20,000.

UPDATE Teacher **SET** T_Salary = 20000 **WHERE** T_sex='female';

If you ignore the **WHERE** clause in the above statement, the query changes the salary of all teachers to $15,000 (See Figure 3.16). Example:

UPDATE Teacher **SET** Teacher.T_Salary = 15000;

Teacher

T_First_Nam	T_Middle_N	T_Last_Nan	T_Sex	T_DOB		T_Salary	T_de	T_IDNO
Haleluya	Biniam	Kiflu	Male	6/13/1988	(1)	$15,000.00	102	Teach/0613/08
Fatuma	Welelaw	Abdi	Female	12/23/1980	(0)	$15,000.00	101	Teach/1010/02
Abebech	Gobena	Ayele	Female	7/7/1985	(0)	$15,000.00	101	Teach/1313/07
Zemikael	O'rero	Samuel	Male	7/30/1978	(0)	$15,000.00	102	Teach/2312/91

Figure 3. 16 SQL Update to Modify Values of Salary of Records of Teacher's Table

Activity 3.11

- Write SQL to update all students' grade_level value to 12.

DELETE command

The DELETE command is used to delete a record or multiple records from the database. **DELETE** command does not remove the table structure, rather it only deletes the data that is currently being held by the table structure.

DELETE command provides the option to mention which records you want to delete based on a condition or delete all records. However, it cannot delete certain fields of a record using the Delete statement. The general syntax of the DELETE statement is given below:

DELETE FROM table_name **WHERE** condition;

Be careful while you use the DELETE command to remove records from tables that are in a one-to-many relationship with another table. For example, if you want to delete one department (one side) already related to other tables, first you have to delete all records in other related tables i.e. in Students, Teacher, and Course Tables (all in the many sides of the relationship). However, deleting a table where the primary key migrates as the foreign key does not have an impact on the Parent table. Therefore, you can delete from the Students table, Course table, and Teacher table if the primary key records are not further related to other tables (See Figure 3.12).

Example: Assume the Teacher named Zemikael does not work anymore in the school. If the teacher is assigned to a course that course should be transferred using the Update statement to another teacher. If not, you can delete the teacher Zemikael only because he is not assigned to offer courses.

```
DELETE FROM Teacher WHERE T_IDNO ='Teach/2312/91';
```

Notes

- When you use the DELETE statement, do not forget to refer to the record by primary key; this is because other fields can have similar values repeated, and you may lose a record.

Deleting Multiple Rows

While putting a conditional clause in the DELETE is optional, it is almost always used – simply because not using it would cause all the records to be deleted from a table, which is a rarely valid need. To delete all records from a table, you can use one of the following commands:

```
DELETE FROM table_name;
```

or

```
DELETE * FROM table_name;
```

Activity 3.12

1. Write SQL command to delete a record from Student table where student_id is 1010/2013.

2. Write SQL statement to delete all records in the Grade table.

3.2.4 Data Query Language – SELECT Command

Brainstorming 3.5

- Have you ever queried Google.com for some information?

Data Query Language (DQL) is a data query language for relational DBMSs. It provides a SELECT command for querying all or subset of records from one or more tables of a database. The SELECT command provides options for filtering

and getting more meaningful results from the database. The level of flexibility that is given by SQL is one of the reasons it has succeeded as a query language in relational DBMSs.

A SELECT command is used to write the SQL statement that contains a complete description of a set of data that you want to obtain from a database. This includes the following:

- What type of data does a table contains?
- How are data from different sources related?
- Which fields or calculations will produce the data?
- Criteria that data must match to be included.
- Whether or not and how to sort the results.

Like a sentence, a SQL statement has clauses. Each clause performs a function for the SQL statement. Some clauses are required in a SELECT statement. The following table lists the most common SQL clauses.

Table 3. 4 Description of SQL SELECT command clauses

SQL clause	What it does	Required
SELECT	Lists the fields that contain data of interest.	Yes
FROM	Lists the tables that contain the fields listed in the SELECT clause.	Yes
WHERE	Specifies field criteria that must be met by each record to be included in the results.	Optional
ORDER BY	Specifies how to sort the results.	Optional

Basic SQL Clauses: SELECT, FROM, and WHERE

This part deals with queries that run on a single table. You can run a SELECT operation on multiple tables in a single statement, too. The syntax to retrieve all records from a table is:

```
SELECT * FROM table_name;
```

The above statement is read as “Select all records from a table”. When the keyword SELECT is followed by the asterisk symbol (*), it means all columns in the table. The asterisk is equivalent to listing all fields in the table. Then write the keyword FROM which should be followed by table name. For example, to select all records of the COURSE table, you write the SELECET statement as follows.

Example:

```
SELECT * FROM COURSE;
```

Figure 3.17 shows the result of running the above SELECT query on school database.

CONTACT_HR	C_NAME	C_DESCRIPTION	C_DEPARTM	C_CODE
2	Accounting Pri	Accounting principle c	104	COM1212
3	Financial Accou	accounting finance	104	COM1312
4	Physics	Fundamentals of Phys	102	NS1211
3	Geography	Ethiopian Geography	101	SS1111

Figure 3. 17 SQL Select Statement Result

Notes

The asterisk (*) in the above SELECT statement can be replaced with all fields in the list and be rewritten as:

```
SELECT CONTACT_HR, C_NAME, C_DESCRIPTION, C_DEPARTMENT,
C_CODE FROM COURSE
```

The output is the same as the one in Figure 3.17. When the table contain all fields, listing all the fields of a table is not important, so it is good practice to use * to specify all columns in the table.

Field name in the SELECT statement should be exactly the same as field name described in the table. Incorrectly spelled field name does not provide the expected output.

Activity 3.13

1. Select all records from the Grade table.
2. Select all records from the Student table.
3. What is the meaning of (*) in the SELECT command.

Selecting a Limited Number of Columns

Most of the time, selecting and displaying all fields of a table may not be your interest; you may want to display a limited set of columns of a table in the output. You can do this by naming the fields explicitly as a comma-separated list (See the syntax below). You can also provide criteria in the WHERE clause if you want to filter records in the columns.

A SELECT SQL statement takes the general form as follows:

```
SELECT field1, [field2]
FROM table
[WHERE criterion];
```

Notes

[WHERE criterion] is an optional clause. It can be used to filter the output that is returned by the FROM clause. The keyword WHERE is followed by an expression that can evaluate to TRUE, FALSE, or UNKNOWN. The expression can be tested using standard mathematical comparison operators (e.g. =, >, >=, <, =<, !=), Boolean operators (e.g. !, AND, OR), or special operators.

String literal (e.g. male) is a sequence of characters enclosed in single quotation marks ('male'). Sometime you may probably find using double quotation marks to enclose string, but the SQL Standard defined to use is single quotation marks.

Example:

The following SELECT query retrieves full name of male students with their sexes from the STUDENT table.

```
SELECT S_Fname, S_MName, S_LName, S_sex
FROM STUDENT
WHERE s_sex = 'male';
```

Figure 3.18 shows the output from the above SELECT query:

Male Students

S_Fname	S_MName	S_LName	S_sex
Yalew	Gobena	Ayele	Male
Abiyot	Zemikael	O'rero	Male

Figure 3.18 SQL SELECT with Limited fields - Results from STUDENT Table

Example: WHERE clause - using comparison in DATETIME

To retrieve records where the values are greater than or equal to the comparison value, retrieve all sales from March 15, 2020 to the present, including sales on that date.

As can be seen in Figure 3.19 below (all records in the SALES table is shown in the left, and sales made on and after March 15, 2020 is shown in the right).

Notes

Comparison in DATETIME determines whether the first value in the expression (i.e. SalesDate) is earlier (if it is <) or later (if it is >) than the second value in the expression. For example, the expression '10-02-2020' < '15-03-2020' as "Is February 10, 2020, earlier than March 15, 2020?" Use >= to consider date values greater than or equal to the comparison value, and <= to consider date values less than or equal to the comparison value. Note also that dates and times are evaluated in their chronological order.

SELECT * FROM SALES

OrderNo	CustomerId	SalesDate	Amount
1005	C511	10-02-2020	$450.00
1008	C213	15-03-2020	$2,000.00
1002	C130	26-01-2020	$2,520.00
1003	C213	30-01-2020	$3,000.00
1006	C130	12-02-2020	$3,599.00
1007	C850	15-03-2020	$7,430.00
1001	C110	26-01-2020	$10,950.00
1011	C213	24-05-2020	$11,200.00
1012	C511	25-05-2020	$13,000.00
1009	C850	20-04-2020	$25,000.00
1004	C110	04-02-2020	$29,000.00
1010	C110	25-04-2020	$30,500.00

SELECT CustomerId, SalesDate, Amount
FROM SALES
WHERE SalesDate >= '15-03-2020'

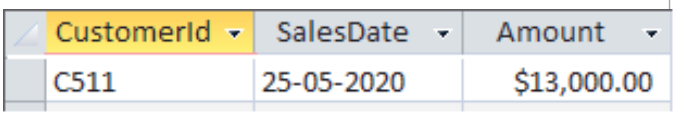
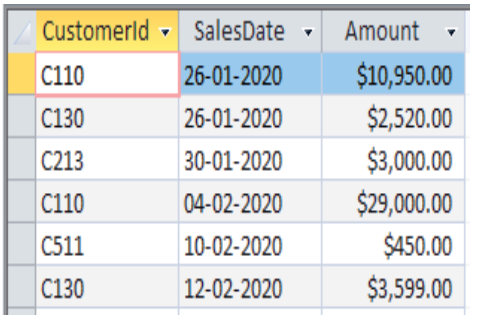
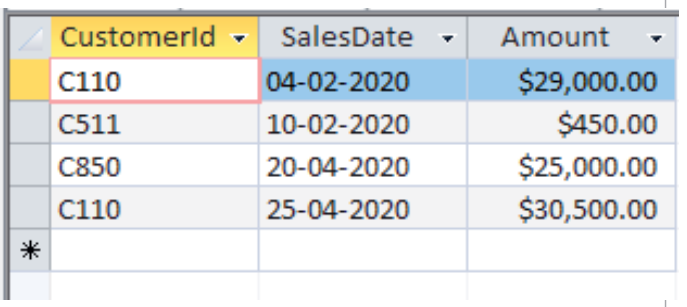
CustomerId	SalesDate	Amount
C850	15-03-2020	$7,430.00
C213	15-03-2020	$2,000.00
C850	20-04-2020	$25,000.00
C110	25-04-2020	$30,500.00
C213	24-05-2020	$11,200.00
C511	25-05-2020	$13,000.00

Figure 3. 19 SALES table with all records (left) and Filtered SALES records (right)

Filter records on multiple criteria: So far, examples of single value comparison has been used to filter records of a table. However, it is common to filter the output of a table based on multiple criteria. In the following example, you see how to combine multiple criteria using AND and OR.

Example: Select records of a table with multiple criteria using AND and OR

If you want to select all sales which are made after May 01, 2020 and its amount is larger than 12000, see Figure 3.20. Figure 3.20 presents additional examples that show the SQL SELECT statement and the output based on SALES table in Figure 3.19.

Query SQL SELECT statement	Result of running the query
SELECT CustomerId, Sales-Date, Amount FROM SALES WHERE SalesDate >=’01-05-2020’ AND Amount > 12000;	 Only one record satisfy the sales date on or later than May 01, 2020 and that sales amount is greater than 12000.
SELECT CustomerId, Sales-Date FROM SALES WHERE SalesDate BE-TWEEN ‘01-01-2020’ AND ‘30-02-2020’;	
SELECT CustomerId, Sales-Date, Amount FROM SALES WHERE Amount <1000 OR Amount > 15000; (Here, the output includes all sales amount less than 1000 or sales amount greater than 15000. There are four rows that meet these criteria based on the SALES table shown above)	

CustomerId	SalesDate	Amount
C511	25-05-2020	$13,000.00

CustomerId	SalesDate	Amount
C110	26-01-2020	$10,950.00
C130	26-01-2020	$2,520.00
C213	30-01-2020	$3,000.00
C110	04-02-2020	$29,000.00
C511	10-02-2020	$450.00
C130	12-02-2020	$3,599.00

CustomerId	SalesDate	Amount
C110	04-02-2020	$29,000.00
C511	10-02-2020	$450.00
C850	20-04-2020	$25,000.00
C110	25-04-2020	$30,500.00

Figure 3. 20 Examples of SELECT statement and the output of running the query.

Notes

The second example in the above figure (BETWEEN … AND) is an example of a range condition, which tests the given value (i.e. SalesDate) against range of values defined by two other values in the expressions (i.e. '01-01-2020' AND '30-02-2020'). The BETWEEN . . . AND defines the range by using the second value in the expression as the start point (i.e. '01-01-2020') and the third value in the expression as the end point ('30-02-2020'). Therefore, both start point and end point are part of the range. Accordingly, a record is included in the output only if the expression of the first value in the expression (i.e. SalesDate) falls within the range specified.

Activity 3.14

Please refer to the Student and the Teacher tables that are used in Activity 3.10.

1. Select all students whose marks are above 50. [Hint: marks>50]
2. Select all female teachers. [Hint: sex = 'F']
3. Select names of students who are grade 12. [Hint: grade = 12]

SELECT Command ORDER BY Clause

Example: SELECT with ORDER BY clause

```
SELECT column1, [column2], …
FROM table_name
ORDER BY [column] [ASC][DESC]
```

You can select records of a table to be ordered in specific field's category. You can order using alphabetical or numerical order. In the syntax above the ORDER BY clause is used to order the output in ascending or descending. The keyword ORDER BY is followed by the field's name i.e., [column], which can be used to order the records in ascending (i.e., if ASC is mentioned or left empty) or descending (i.e., if DESC is mentioned). ASC stands for ascending and DESC stands for descending. Ascending is the default order. Writing the query in Figure 3.21 retrieves the full names of the teachers with their salary in ascending order (See Figure 3.21). T_Salary is followed by nothing, and the output is that salary is in ascending order (See table result in Figure 3.22).

```
SELECT T_First_Name, T_Middle_Name, T_Sex, T_Salary

FROM Teacher

ORDER BY T_Salary
```

T_First_Nam	T_Middle_N	T_Sex	T_Salary
Fatuma	Welelaw	Female	$12,000.00
Abebech	Gobena	Female	$14,000.00
Zemikael	O'reo	Male	$15,000.00
Hanfere	Aligaz	Male	$16,000.00

Figure 3. 21 SELECT with ORDERE BY on the Teachers table – SQL statement and Output

To get salary in descending order, use the ORDER T_Salary DESC as follows:

```
SELECT T_First_Name, T_Middle_Name, T_Sex, T_Salary

FROM TEACHER

ORDER BY T_Salary DESC
```

T_First_Nam	T_Middle_N	T_Sex	T_Salary
Hanfere	Aligaz	Male	$16,000.00
Zemikael	O'reo	Male	$15,000.00
Abebech	Gobena	Female	$14,000.00
Fatuma	Welelaw	Female	$12,000.00

Figure 3. 22 Teachers table records ordered by salary in descending order

Notes

- Did you know that the order of records and columns within a table has no importance in database? However, you can use the ORDER BY clause to order the output in ascending or descending order.

Activity 3.15

1. Select all teachers who are specialized in Mathematics and order them by their names.
2. Select all records in the Teacher table and order them by sex.
3. Perform the following activities based on the table below
   - Select patients and order by Temperature in descending order.
   - Select patients and order by VisitDate in ascending order.

PatientId	VisitDate	Physician	Temperature	BodyPressure
89101	10-06-2021	Amare	38	120/70
89009	13-02-2022	Negesti	37.5	120/70
89203	15-02-2021	Murad	39	150/60
90555	04-10-2021	Azeb	38.5	130/80
89777	29-01-2022	Chala	39	160/70

Selecting records from Two Tables

The SELECT command can be used to select records from two tables. While selecting records from two tables, the records in the two tables must have some link; otherwise, the resulting set does not provide relevant information. The general syntax of the SELECT statement that is to select records from two tables is given below.

```
SELECT table1.column1, [table1.column2], table2.column1, [table2.column2]
FROM table1, table2
WHERE table1.column1 = table2.column2
```

Notes

Unlike the earlier syntax, the column is described along with table that contains the field. For example, table1.column1, refers to column1 in table1. Describing column name along with their table is important when you are selecting records from two tables. This practice avoids naming conflict among similar column name found in two tables.

Example: Your school director wants to get courses offered in the first semester. We know that the course table maintains course details with Course id as a primary key. This table alone cannot provide the information that is requested by the director. Because courses offered in a semester is maintained in another table (CourseOffering). The Course id (i.e., a primary key in the COURSE table) is a foreign key in the CourseOffering table. This key is used to link the two tables. Therefore, to provide full information that is requested by the school director, part of the information can be extracted from Course as well as CourseOffering tables.

```
SELECT COURSE.Course_id, COURSE.Course_name, COURSE.Grade,
CourseOffering.Semester

FROM COURSE, CourseOffering

WHERE COURSE.Course_id = CourseOffering.Course_code;
```

Table 3. 5 Selecting records from two tables - Course and CourseOffering

Course table

Course_ id	Course_ name	Grade	Contact_Hr
NS103	Physics	11	4
NS104	Biology	11	4
NS102	Biology	10	4
NS105	English	11	5
NS101	Chemistry	9	4

CourseOffering table

Course_ code	Teacher_id	Sec- tion	Semes- ter
NS104	2010/2010	A	I
NS104	2010/2020	B	I
NS103	2015/2011	A	II
NS102	2010/2010	C	II
NS105	2002/2020	A	I

The output of the query

Course_id	Course_name	Grade		Semester
NS103	Physics	11		II
NS104	Biology	11		I
NS104	Biology	11		I
NS102	Biology	10		II
NS105	English	11		I

Example 2: The output in Table 3.5 can be further filtered by adding the criterion "**Semester= 'I'** " as follows:

```
SELECT COURSE.Course_id, COURSE.Course_name, COURSE.Grade,
CourseOffering.Semester

FROM COURSE, CourseOffering

WHERE COURSE.Course_id = CourseOffering.Course_code AND Semester='I';
```

Therefore, the output of the query includes only Semester I (as shown in Table 3.6)

Table 3. 6 Example query statement with multiple criteria

Course_id	Course_name	Grade	Semester
NS104	Biology	11	I
NS104	Biology	11	I
NS105	English	11	I

Unit Summary

In this unit, you have learnt about:

- concepts of Database Management Systems(DBMSs).
- Relational Database Management Systems
- Structured Query Language(SQL)
- DDL, DML and DQL

Key Terms

There are many advantages of having databases, some of the major ones are sharing, security, less data redundancy, and data integrity.

DBMS is a computerized system that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.

Databases are composed of many objects, but the major objects that exist in MS Access are tables, queries, forms, and reports

SQL is a special purpose query language meant for interacting with relational databases. Understanding how SQL works can help you create better queries, and make it easier for you to understand how to fix a query that is not returning the results that you want.

SQL is a language for interacting with databases. It consists of a number of commands with further options to allow you to carry out your operations with a database.

All **RDBMS** commands are written using the SQL statements.

Data Definition Language (DDL) which includes CREATE, ALTER and DROP. These commands allow you to create or modify your database structure.

Data Manipulation Language (DML) contains INSERT, UPDATE, and DELETE. These commands are used to manipulate data stored inside your database.

Data Query Language (DQL) includes SELECT that is used for querying or selecting a subset of data from a database.

Review Questions

Part I: Choose the correct answer from the given alternatives.

1. Which one of the following is generally used for performing tasks like creating the structure of the relations?

   A. Query
   B. Relational Schema
   C. DML(Data Manipulation Language)
   D. DDL(Data Definition Language)

2. Which one of the following provides the ability to insert tuples into, delete tuples from, and modify tuples in the database?

   A. Query
   B. Relational Schema
   C. DML(Data Manipulation Language)
   D. DDL(Data Definition Language)

3. A Database Management System is a type of_____ software.

   A. system
   B. application
   C. general
   D. Both A and C

4. Which one of the following is a type of Data Manipulation Command?

   A. CREATE
   B. DELETE
   C. ALTER
   D. All of the above

5. Which of the following command is a type of Data Definition language command?

   A. CREATE
   B. UPDATE
   C. DELETE
   D. MERGE

6. Which one of the following commands is used to remove the existing row in a table

   A. DELETE
   B. UPDATE
   C. INSERT
   D. None of the above

7. Which language is used by most DBMS for helping its users access data?

   A. HLL
   B. Query language
   C. SQL
   D. 4 GL

8. Microsoft Access is a_____.

   A. Network Database Model
   B. RDBMS
   C. ORDBMS
   D. OODBMS

Based on the following table, answer the questions that follows.

ACCOUNT		
NAME	**PASSWORD**	**EMAIL**
Almaz_bekana	Pass@MY123	Almaz_bekana@mymail.com
Asnakech worku	MYPass@123	Asnakech_worku@mymail.com
Diriba Dilnesa	PassMY@123	Diriba_Dilnesa@mymail.com
Sofonias Alebel	MY@123pass	Sofonias_Alebel@mymail.com
Muna Mussa	Pass@123MY	Muna_Mussa@mymail.com

9. Based on the given table and the datatype on the data filled, which query creates the above table?

   A. CREATE TABLE ACCOUNT (NAME char (30), PASSWORD int, EMAIL char (30));
   B. CREATE TABLE ACCOUNT (NAME char (10), PASSWORD char (10), EMAIL char (10));
   C. CREATE TABLE ACCOUNT (NAME char (30), PASSWORD char (15), EMAIL char (30));
   D. Create ACCOUNT (NAME char (30), PASSWORD char (15), EMAIL char (10));

10. Which of the following is used to define a header for HTML tables?

   A. INSERT ACCOUNT values (‘Abebe’, ‘mypassword’, abebe@mymail.com)
   B. INSERT INTO ACCOUNT (Name, Password) values (‘Abebe’, ‘mypassword’)
   C. INSERT INTO ACCOUNT values (‘Abebe’, ‘abebe@mymail.com’)
   D. Insert values to theACCOUNT table (‘Abebe’, ‘mypassword’)

11. Which of the following is used to define a header for HTML tables?

A. Update ACCOUNT password='mypassword@123';
B. Update ACCOUNT set password='mypassword@123';
C. Update table ACCOUNT set password='mypassword@123';
D. None of the above

12. Which of the following is used to define a header for HTML tables?

A. Delete from ACCOUNT
B. Delete Hailemariam Mamo from ACCOUNT
C. Delete from ACCOUNT where name = 'Hailemariam Mamo';
D. Delete record 'Hailemariam Mamo' from ACCOUNT;

13. Which one of the following commands does not enable you to select data from the table ACCOUNT given above?

A. SELECT * from ACCOUNT
B. SELECT name, password from ACCOUNT where password = 'MY@123pass'
C. SELECT * from ACCOUNT where Password like '*123'
D. SELECT All from Account

Part II: Code Writing

Identify a data management problem which can be addressed by designing a database solution.

Hints: You may work on a patient record management system, a farmer record management system, a school's record management system, a business customer management system, a library management system, a hotel management system or any other problem in your surrounding that can be addressed through the design and implementation of database system.

Instructions:

A. Analyze the problem you want to work on and identify three to five objects or entities about which you want to maintain data.

Hint: Patient, Physician, Treatment, etc if you work on patient management system in Clinic; Guest, BedRooms, CheckIn, etc if you work on hotel management system.

B. Design the database structure.

Hint: Develop the ERD (Entity Relationship Diagram), identify attributes, **and** create link (when appropriate).

C. Create the database and tables in it.

Hint: You may find informative name for your database from the problem you considered. E.g. 'PatientRecordDatabase', 'BookManagmentDatabase'.

D. Review your tables and modify tables, fields, data type, or table link with other tables if needed. **Hint**: Use **ALTER** command.

E. Add three to five records in each table that have been created so far.

Hint: Use INSERT INTO command to add new records in a table.

F. Update at least two records in one of your tables.

Hint: Use UPDATE command to update records in a table.

G. Write five queries based on the following criteria.

- Select all records (e.g. select all patients, or doctors).
- Select records based on a single criterion

Hint: WHERE patientAdmitedDate earlier than January 1, 2022

- Select records based on multiple records.
- Hint: WHERE physicianAge < 45 AND Salary = 7000

UNIT 4

WEB AUTHORING

Learning Outcomes

At the end of this unit, students will be able to:

- Explain how to use an extensible markup language(XML)
- Analyse an XML document
- Describe basic XML tags
- Apply the syntax rules of XML documents
- Create simple XML documents
- Compare and contrast hypertext markup language(HTML) vs. XML
- Understand how to publish a website

Unit Overview

HTML, which stands for Hyper Text Markup Language, is the standard markup language for creating web pages. HTML consists of a series of elements that are used to describe the structure of a web page, such as <html>, <head>, <title>, <body>, <table>, <image>,<a>, etc. These elements tell the browser how to structure and display the content. This means, the HTML elements label pieces of content such as “My Heading”, “Paragraph starts here”, “Link to about page”, etc. You have learned the basics of HTML and web design in Web Authoring in Grade 11 IT subject.

This unit introduces another widely applicable markup language called XML. The unit covers an overview of XML, its structure, and the differences between XML and HTML. Finally, website publishing is discussed.

4.1. Introduction to XML

Brainstorming 4.1

- What comes to your mind when you hear the term 'markup language'?

The **eX**tensible **M**arkup **L**anguage (XML) is a markup language like HTML. A markup language is a computer language that uses tags enclosed with less than (<) and greater than (>) symbols to define elements within a document. When the file is processed by a suitable application, the tags are used to control the structure or presentation of the data contained in the file. Any text that appears within one of these tags is considered part of the markup language. Markup files contain standard words, rather than typical programming syntax.

Activity 4.1

- Describe one fact about XML.

XML is a way of applying structure to a web page. XML provides a standard open format and mechanisms for structuring a document so that it can be exchanged and manipulated. XML complements your HTML knowledge by allowing you to structure your data by marking up the text and data to define the data content. XML is used for storing structured data, rather than formatting information on a page. While HTML documents use predefined tags (such as " <head>", "<title> "), XML files use custom tags to define elements.

Microsoft Rich Text Format (RTF), Adobe Portable Document Format (PDF), and HTML are types of markup languages that provide presentational markup. They are powerful solutions to the problem of displaying information. Their common limitation is that they describe how the data looks, but they do not give any information about what it is.

Like HTML, XML uses tags to "mark-up" data content. Unlike HTML, in XML you define your own tags that meet the exact needs of your document. The custom

tags make your data easier to organize and more searchable. For example, a student might describe the book she reads on vacation time like this:

```
<books>
        <book>
        <title>An Introduction to XML and Web Technologies </title>
        <author> by Anders Moller, Michael Schwartzbach</author>
        <type>Programming Languages</type>
        </book>
</books>
```

XML does not change the way your web pages looks; instead, it changes the way the documents are read and filed and stored. Therefore, XML is used to describe the structure of a document rather than the way it is presented.

The two areas in which XML is useful are structuring data for storage, where a relational database (See Unit Three about database) is inappropriate, and the presentation of web pages. For example, a system is handling small quantities of data, or if the data lacks a relational structure, programmers usually prefer to create their own data formats, i.e. XML.

Activity 4.2

1. What tags are used by XML? How are they different from tags used in HTML?
2. Can HTML replaces XML? Explain your reason.
3. Describe two areas where XML is important.

4.1.1 Elements of XML Documents

XML document must contain one root element that is the parent of all other elements, for example "<people>" in Figure 4.1). The best way to learn what makes up an XML document is by starting from a simple example. The following is a complete XML document that lists the names of two known athletes.

XML document structure	Example of an XML document
<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE people[<!ELEMENT people (person+)> <!ELEMENT person (name)> <!ELEMENT name (first, last)> <!ELEMENT first (#PCDATA)> <!ELEMENT last (#PCDATA)>]>	<people> <person> <name> <first>Haile </first> <last>Gebrselassie</last> </name> </person> <person> <name> <first>Derartu</first> <last>Tulu</last> </name> </person> </people>

Figure 4. 1 XML Document Structure (left) and Example (right)

The above XML lets you name parts of the document such as the first names and the last names of two of our legendary athletes. You can name anything that is relevant to the document you want to produce. All that matters is that you follow the basic rules for creating tags. The above markup does not follow the basic rules (predefined tags), rather you define your tags. Figure 4.2 highlights the various elements of the sample document.

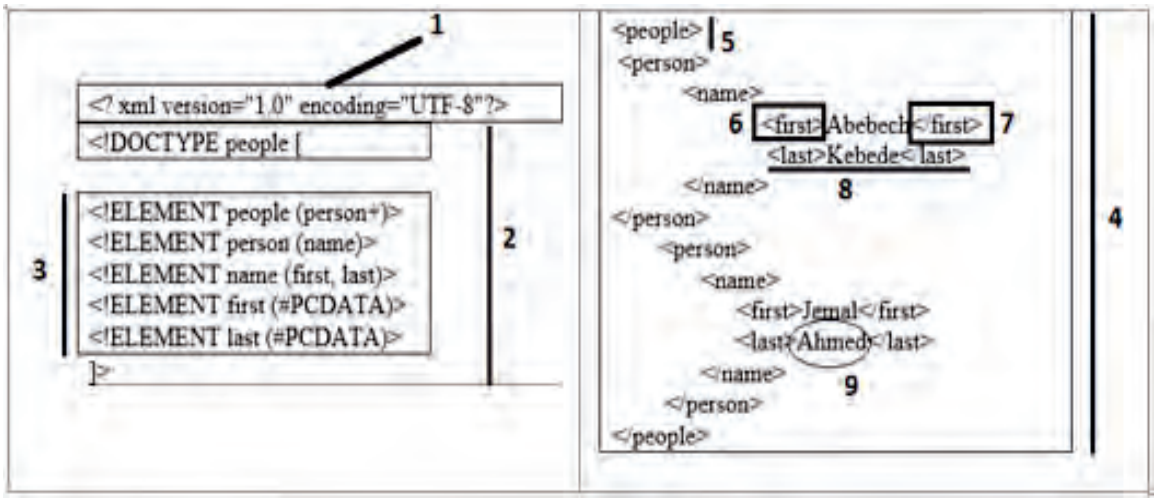


Figure 4. 2 Various Elements of the XML document

The numbers shown in Figure 4.2 are interpreted as follows:

1. **XML declaration:** describes the general characteristics of the document, such as XML document, the version of the XML, and the encoding character it uses. XML documents usually begin with the XML declaration statement called the processing instructions which provide information on how the XML file should be processed. E.g. <?XML version= "1.0" encoding="UTF-8"?> The processing instruction statement uses the encoding property to specify the encoding scheme used to create the XML file. Encoding is the process of converting Unicode character into their binary equivalent representation depending on the type of encoding('UTF-8' or 'UTF-16').
2. **Document Type Declaration (DTD):** describes the structure of the document in terms of which elements it may contain, along with any restrictions it may have. In other words, it describes the root. The above example is about **people**. The document 'people' is described with five elements. These are described below.

   E.g. <!DOCTYPE people […]>
3. **Internal DTD subset:** a DTD is internal if the elements are declared within the same XML file. In the following example, internal declarations that are local to the XML document are used.

   <!ELEMENT people (person+)>

   <!ELEMENT person (name)>

   <!ELEMENT name (first, last)>

<!ELEMENT first(#PCDATA)>

<!ELEMENT last(#PCDATA)>

Notes

- **PCDATA** – stands for **P**arsed **C**haracter **DATA**. Character data is a text that is found between the start tag and the end tag of an XML element. E.g. Abebech in Figure 4.2.

4. **XML information set or Content:** this represents the XML document's content—the information that the document conveys. Content refers to the information that is represented by the elements of an XML document. See the example in Figure 4.3.

```
<people>
        <person>
                <name>
                                <first>Abebech</first>
                                 <last>Kebede</last>
                </name
</person>
<person>
        <name>
                                 <first>Jemal</first>
                                <last>Ahmed</last>
        </name>
</person>
</people>
```

Figure 4. 3 XML document content.

5. **Root element:** This encloses all of the information. An XML document can have only one root element. Therefore, "<people>" is the root of this XML document.
6. **Start tag:** XML elements have a start and an end tag—the start tag provides the name of the XML element. E.g. <first>
7. **End tag:** The name of the end tag must exactly match the name of the start tag.

E.g. <people> with </people>, <person> with </person>, <first> with </first>

8. **XML element:** The start and the end tags are collectively referred to as an XML elements. Elements are the basic units that are used to identify and describe the data in XML. They are the building blocks of an XML document. E.g. <last>Kebede</last>
9. **Data:** XML elements can contain data between the start and the end tags. An XML document represents information using a hierarchy. That means, it begins with a root element (e.g. people), which contains sub-elements (e.g. person) which in turn can contain other sub-elements(e.g. name), data (e.g. Kebede), or both. E.g. Abebech, Kebede, Jemal, and Ahmed are data.

Besides the above elements, attributes and comments are also part of XML documents.

Attribute: Like HTML, XML elements can contain attributes. An attribute provides additional information about the elements for which it is declared. It consists of a name-value pair. In the following example, the attribute name is **personid** and the value is "101". The attribute value should be quoted: single or double quotes can be used.

E.g. <name personid = **"101"**>Ubang </name>

Comment: This is a kind of note or statement that is used to describe the XML code. Comments can provide documentation information about the XML file or the application to which the file belongs. A comment is ignored by the XML parser(or a program that interpretes XML instruction) during code execution.

The syntax for a comment is: <!-- This is a comment -->

We can see the whole hierarchy of the above markup in an upside-down tree structure, as shown in Figure 4.4.

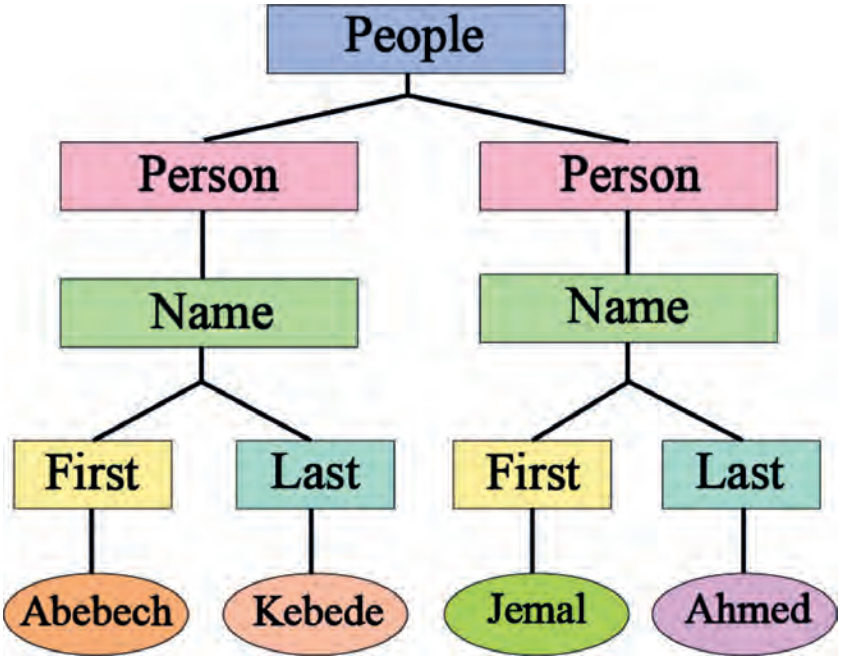


Figure 4. 4 Tree Structure of Person XML document

Although XML is designed so that people can read it, it is not intended to create a finished document. In other words, you can not open up just any XML-tagged document in a browser and expect it to be formatted well. XML is meant to hold content so that when the document is combined with other resources, such as a style sheet which renders a web page, it becomes a finished product.

Activity 4.3

1. Describe elements of XML documents and their purposes.
2. Write XML that describes a food menu. Each type of food has a description which contains the name of the food and its price. Save it as test.xml.

Hint: root – food_menu, DTD – it is about food, content – food description (name and price).

4.1.2 Creating XML Documents

There are a few ways of opening an XML file directly. You can open and edit XML files with any **text editor**, view them with any **web browser**, or use a website that lets you view, edit, and even convert them to other formats. You can also use

applications such as “oxygen” or “XML Notepad” to see your files’ structures. In this section, we use “XML Notepad” for our demonstration.

For example, after saving the XML code given above as an example with the *.xml extension in any text editor, when you open the file with XML Notepad, it looks as shown in Figure 4.5. It just does nothing because XML is just information wrapped in tags. A piece of program should be written to send, receive, store, or display it.

When you view your XML document in a browser, most browsers display an XML document with color-coded elements (See Figure 4.5 below comparing the document Mozilla and Internet Explorer browsers). Often a plus (+) or minus (-) sign to the left of the elements can be clicked to expand or collapse the element structure. To view XML source code, try to select “View Page Source” or “View Source” from the browser menu you use.

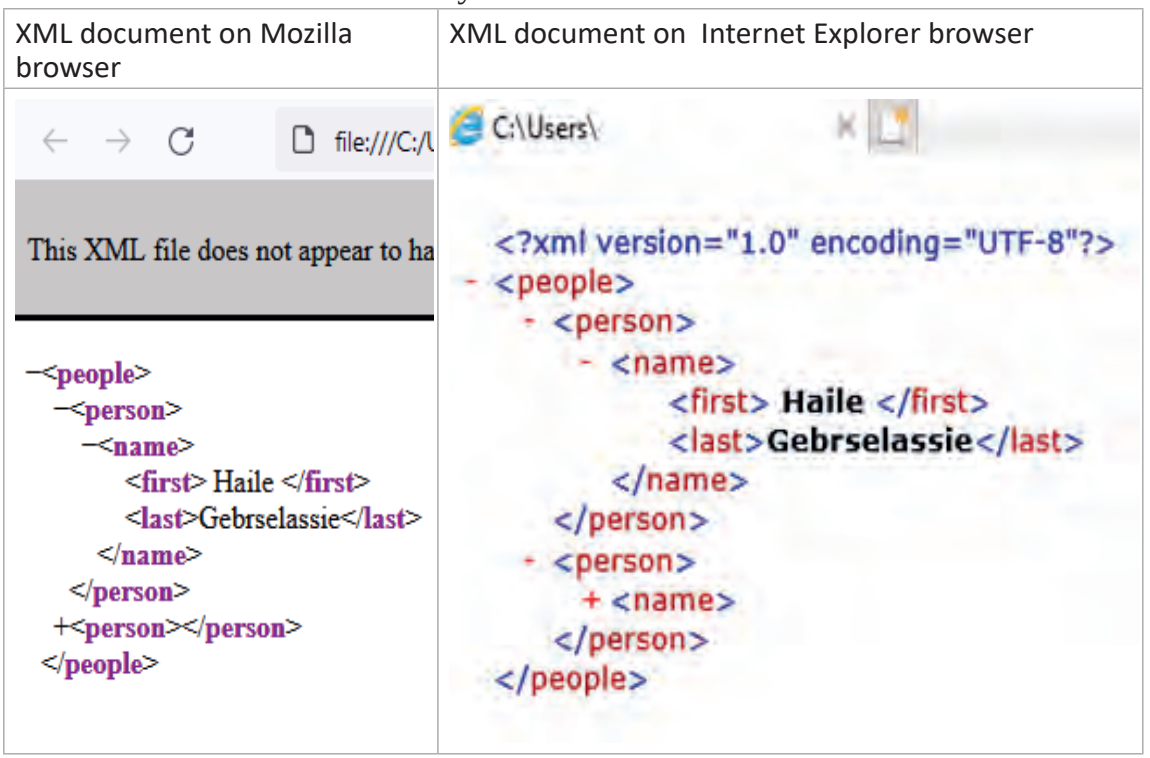
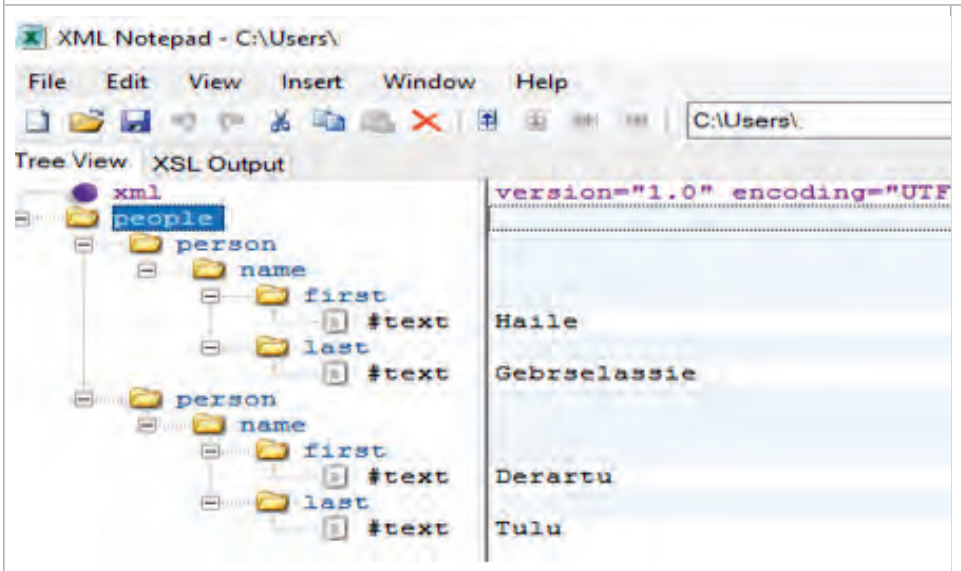


Figure 4. 5 XML Document on Browsers – Mozilla and Internet Explorer

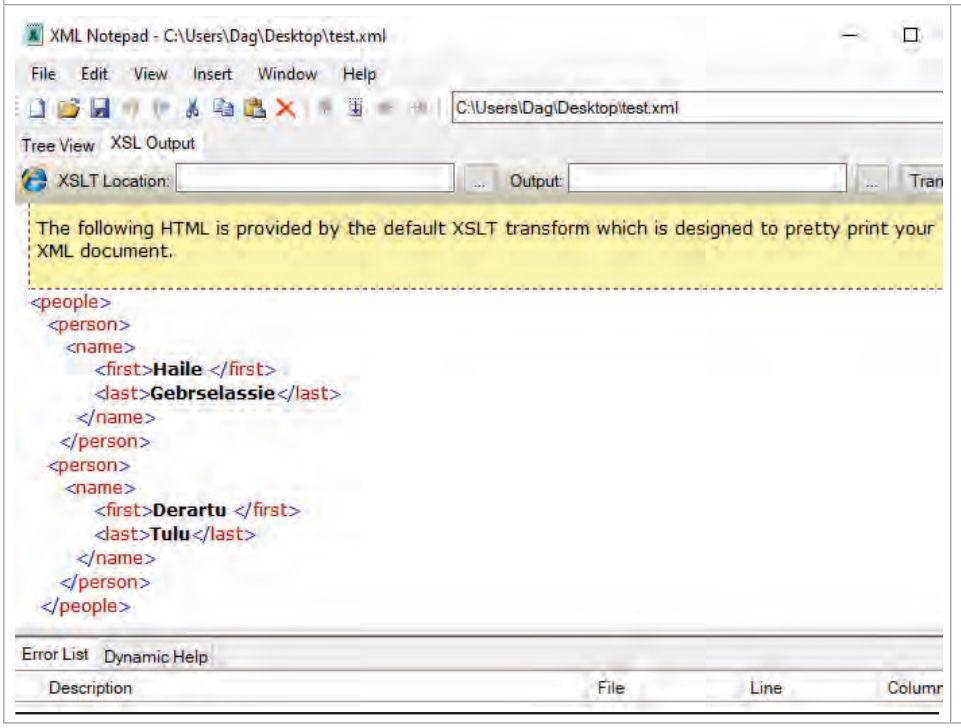
Figure 4.6 below demonstrates a view of an XML document using XML Notepad editor.

XML document viewed on XML Notepad – Tree View



Tree View on XML Notepad

XML documents viewed on XML Notepad – XSL Output view



XSL Output on XML Notepad

Figure 4. 6 XML document view on XML Notepad

Notes

- XSL is a style sheet for XML as CSS is a style sheet for HTML. XSL transform and render XML document (See Figure 4.6 above).

Activity 4.4

1. Write the above XML on any text editor and save it as test.xml. Then open the test.xml using at least two browsers and note what you have observed.
2. Use the file created in Activity 4.3 Question 2 and open the test.xml using XML Notepad editor and see the result – in Tree View and XSL Output.

Hint: Don't forget to add XML declaration statement at the beginning.<?xml version="1.0" encoding="UTF-8"?>

Syntax rules to be remembered while working on XML:

1. You should have one **root**: XML documents must contain one element that is the **parent** of all other elements.
   A. <A><X> . . . </X><Y>…</Y></A></X> is wrong
   because the root tag starts with <A> and ends with a different tag </X>
   B. <A><X> . . . </X><Y>. . . </Y>**</A>**is correct
   because the root tag starts with <A> and ends with the same tag </A>
2. XML elements are **case sensitive**
   A. <**n**ame>Ibrahim</**N**ame> is wrong
   B. <**NAME**>Ibrahim</**N**ame> is wrong
   C. <Name>Ibrahim</Name> is correct
3. All elements should not overlap, and they **must** properly be nested within each other.
   A. <**Name**><Student>Almaz</**Name**></Student> is wrong
   B. <Student><**Name**>Almaz</Student></**Name**> is wrong
   C. <Student><**Name**>Almaz</**Name**></Student> is correct
4. XML attribute values must always be **quoted.** The XML elements can have attributes in name-value pairs which are similar to HTML.

```
<note date="12/11/2007">
  <to>Gebre</to>
  <from>Almaz</from>
</note>
```

5. **Comments in XML:** The syntax for writing comments in XML is similar to that of HTML
   E.g. <!-- This is a comment -->

6. XML does not truncate multiple white-spaces. In HTML, multiple white-spaces are truncated to a single white-space.

XML	HTML
Hello AbebechGobena	Hello AbebechGobena

7. **Entity References:** In XML, some characters have special meaning. For example, a character like "<" inside an XML element will generate an error. This is because the XML parser interprets "<" as the start of a new XML element. See the example below:

<Message>if salary <1000 then</Message>	This generates an XML error:

To avoid this error, replace the **"<"** character with an **entity reference** as:

<Message>if salary **<** 1000 then</Message>	This is correct

There are 5 pre-defined entity references in XML. See these references in Table 4.1 below.

Table 4. 1 Entity References in XML

<	<	less than
>	>	greater than
&	&	ampersand
'	‘	apostrophe
"	“	quotation mark

Notes

- Only the characters "<" and "&" are strictly illegal in XML. The greater than character is legal, but it is a good habit to replace it. Therefore, XML documents that conform to the syntax rules above are said to be "**Well Formed**" XML documents.

Activity 4.5

Examine the following XML code and rewrite it if the code is wrong.

XML code to analyze	Corrected XML code
<note><to>Lucy</to><from>Hadar</note></from>	
<student id=1010><name>Edossa</name></student>	
<birth_date>Date of Birth</Birth_date>	
<compare>10>5</compare>	
<FullName> Robso Amanueal </fullname>	

Element Declarations: XML documents consist of three things: ***elements***, entities, and ***control information***. Elements are surrounded by tags like HTML. The content of the document has a structure imposed by the rules of XML although this structure is quite loose. Each document has a single root element that contains all of the other markups. You have already learned this point in HTML where all documents are enclosed inside <html></html> tags. The document is then composed of several sections, each of which is enclosed between tags. The sections themselves are also elements.

An XML element is everything from the **start tag** to the **end tag's** of the element. An element without content is said to be **empty** element. An empty element can be described in one of the following options.

<element></element>	<element />

Naming rules of XML elements:
• Element names are case-sensitive. • Element names must start with a letter or underscore. • Element names cannot start with the letters XML. • Element names can contain letters, digits, hyphens, underscores, and periods. • Element names cannot contain spaces.

Any name can be used, but no words are reserved (except XML).

An element can contain:	Example of XML
• Text • Attributes • Other Elements, or • A mix of the above	<Bookstore> <Book category="children"> <title> Stories and Parables</title> <author>Kebede Michael </author> <year>2007</year> <price>Birr 150.00 </price> </Book> </Bookstore>

Figure 4. 7 XML Elements vs Examples

In the Figure 4.7 above,

- <title>, <author>, <year>, and <price> have **text content** because they contain text. E.g. Kebede Michael.
- <Bookstore> and <Book> have **element contents** because they contain elements.
- <Book> has an **attribute**. E.g. category=”children”.

Attribute Declarations: An XML element, like HTML, can have attributes. The element’s attributes define data related to a specific element. Attributes provide additional information about the element. An attribute is defined as key-value pair: the key is the name (e.g. gender) and the attribute has values that must always be **quoted (e.g. “female”)**. Either single quotation marks (‘ ’) or double quotation marks (“ ”) can be used. For example, a person element’s gender can be written like this:

<person gender= “female”>or <person gender= ‘male’>

If the value of the attribute itself contains double quotation marks, use single quotation marks, as shown in this example:

<person name= ‘Tilahun Gessesse “KomeLimerkish(ቆሜልመርቅሽ)”’ >

XML Elements vs. Attributes

<person gender=”female”> <firstname>Anna</firstname> <lastname>Smith</lastname> </person>	<person> <gender>female</gender> <firstname>Anna</firstname> <lastname>Smith</lastname> </person>

Figure 4. 8 XML Elements vs. Attributes

In the first example of Figure 4.8, gender is an attribute. In the next example, gender is an element. Both examples provide the same information. There are no rules about when to use attributes or elements in XML.

Entity Declarations: An entity is a declaration of a name that can be used in an XML in place of content or markup. Entities must be declared before they are used. All entities are declared with the “ENTITY” declaration. However, the exact format of the declaration distinguishes between internal, external, and parameter

entities. The internal (parsed) entity declaration will be discussed for our purpose. The purpose of internal parsed entities is generally to refer to text. It refers to data that an XML processor has to parse.

Syntax for creating entity declaration	Example
<!ENTITY name “value”>	<!ENTITY publisher “Berhanena Selam Printing”>

Figure 4. 9 Entity Declaration

There are two parts to entities. The first is an entity declaration of a name that tie the replacement content. The above example declares an entity of the name ‘publisher’. The name ‘publisher’ is associated with the content of “Berhanena Selam Printing”. The second is using the name defined in the entity declaration subsequently in the XML as shown below. When the entity name is used in XML, it is called an entity reference.

<author>Addis Alemayehu</author>

<pubinfo>Published by **&publisher**;</pubinfo>

When the parser subsequently encounters an ampersand symbol (i.e. &) and semicolon (i.e. ;) which identify a general entity reference, the parser looks up that name in an entity declaration table (See Figure 4.9). In our case, the reference, **&publisher;** is replaced by the content that it represents. Using the following (left column) XML, expanding the entity reference, and replacing the **&publisher;** with the “Berhanena Selam Printing” content gives the following expanded XML.

XML document with internal DTD
<<?xml version=”1.0”?>
<!DOCTYPE book [
<!ENTITY publisher “Berhanena Selam Printing”>
]>
<book>
<author>Addis Alemayehu</author>
<pubinfo>Published by &publisher;</pubinfo>
</book>

Output on browser

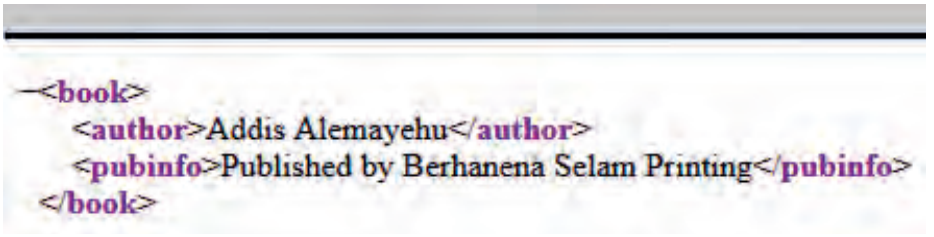


Figure 4. 10 XML Document with DTD and Output on Browser

Document Type Definition (DTD)

A Document Type Definition (DTD) defines the structure attributes of a document and the legal elements of an XML document. It can be used by an application to verify that XML data is valid. A DTD declared inside the XML file must be wrapped inside the <!DOCTYPE> definition. See Figure 4.11 below.

Example of an XML document with internal DTD

```
<?xml version="1.0"?>
<!DOCTYPE album [
<!ELEMENT album (singer, name, release, language)>
<!ELEMENT singer (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT release (#PCDATA)>
<!ELEMENT language (#PCDATA)>
]>
<album>
<singer>Ali Birra</singer>
<name>Ammalele</name>
<release>2012</release>
<language>Oromiffa</language>
</album>
```

The interpretation of the above DTD is:

- **!DOCTYPE album -** defines the root element of the document album
- **!ELEMENT album -** defines the album element which must contain four elements: “singer, name, release, language”
- **!ELEMENT singer -** defines singer element type “#PCDATA”
- **!ELEMENT name -** defines name element type “#PCDATA”
- **!ELEMENT release -** defines release element type “#PCDATA”
- **!ELEMENT language -** defines language element type “#PCDATA”
- **PCDATA** - means parsed character data. Think of character data as the text that is found between the start tag and the end tag of an XML element.

Figure 4. 11 Document Type Definition

Activity 4.6

- Write DTD for the following XML and test the output on a browser.

```
<!ENTITY author “Author: Addis Ababa University.”>
<!ENTITY copyright “Copyright: Ministry of Education.”>
<textbook>&author;&copyright;</textbook>
```

4.2. HTML vs. XML

Brainstorming 4.2

- Compare the features of HTML and XML.

HTML is the markup language that helps you to create and design web content. It has a variety of tags and attributes for defining the layout and structure of the web document. It is designed to display data in a formatted manner. An HTML document has the extension .htm or .html.

XML is a markup language that is designed to store data. It is popularly used for the transfer of data. It is case sensitive. XML offers you the ability to define markup elements and generate customized markup language. The basic unit in XML is known as an element, and the extension of an XML file is .xml

Table 4.2 Compares common features of HTML and XML.

Parameter	**XML**	**HTML**
Type of language	XML is a framework for specifying markup languages.	HTML is a predefined markup language.
Structural details	They are provided.	They are not provided.
Purpose	Transfer of data.	Display / Presentation of data
Nesting	It Should be done appropriately.	It does not have any effect on the code.
Driven by	XML is content driven.	HTML is format driven.
Size	Documents are mostly lengthy in size, especially when an element-centric approach is used in formatting.	The syntax is very brief and yields formatted text.
Learning curve	It is very hard as you need to learn technologies like XPath, XML Schema, DOM, etc.	It is a simple technology stack that is familiar to developers.
Coding Errors	No coding errors are allowed.	Small errors are ignored.
Extension	.xml	.html or .htm
	E.G. Page1**.xml**	E.G. Page1**.html**
Whitespace	White spaces can be used in your code.	White spaces cannot be used in your code.
Output	<Name> Motherland Ethiopia</Name>	Motherland Ethiopia
	Motherland Ethiopia	**Motherland Ethiopia**

Parameter	XML	HTML
Language type	It is case sensitive.	It is case insensitive.
	<**N**ame>Lucy</**N**ame>	<**STR**ong>Lucy<**str**ONG>
Tag	Tags are defined as per the need of the programmer.	It has its own predefined tags.
	E.G. <book><Author><Title>	E.G. <body><b><i>
End of tags	The closing tag is essential in a well-formed XML document.	The closing tag is not always required.
	<Person><student><Name>Your name </Name></student>></Person>	<body>**<P>** This is paragraph </body>
Quotes	Quotation marks are required around XML attribute values.	Quotation marks are not required for the values of attributes.
	<Department><number type=**"int"**> 101 </number></Department>	<body bgcolor=**#00ff00**><p> This is paragraph </body>

Activity 4.7

- Discuss the similarities and differences between HTML and XML.

4.2.1 Advantages and Disadvantages of XML

Some of the advantages and disadvantages of XML are listed below.

Table 4. 3 The Advantages and the Disadvantages of XML

Advantages of using XML	Disadvantages of using XML
• XML makes documents transportable across systems and applications. With the help of XML, you can exchange data quickly between different platforms. • It separates the data from HTML. • It simplifies the platform exchange process.	• XML requires a processing application • The XML syntax is very similar to other alternative 'text-based' data transmission formats which are sometimes confusing • No intrinsic data type support • The XML syntax is redundant • It does not allow users to create their tags.

Activity 4.8

- Discuss the advantages and disadvantages of XML with a partner.

4.3. Publishing Website

Brainstorming 4.3

- What do you know about website publishing?

You have learned that the purpose of XML is for storing structured data, not formatting or styling a document. That is why you see on top of the browser **"This XML file does not appear to have any style information associated with it"** when you open your XML document (See Figure 4.5 above).

XML is just information wrapped in tags. So it is now time to write a piece of program to display our XML document. In the following discussion, you will learn how you put all of the XML document, web designing, and publishing concepts that you have learnt in Grade 11 together to develop further skills in website publishing.

Step-by-Step design of XML document and publishing

So far, it has been discussed that HTML tags are not understood by XML coding. If that is the case, we can have our own customized tag names for the specific environment rather than predefined HTML tags. Let us take a 'college' environment and organize the contents of the college documents using an XML structure.

To start your coding, you should write an XML declaration statement at the beginning to indicate that it is an XML language by using **<? Xml version =”1.0” encoding=”UTF-8”?>,** as shown in the example below.

Instead of <**html**>, we can start our root to be <**College**>. This shows that the initial data or root for our XML is going to be college:

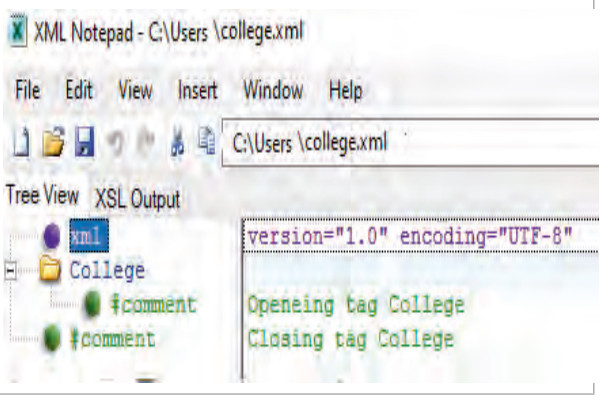
XML document Code	XML document view on XML Notepad
<?xml version =”1.0” encoding=”UTF-8”?> <**College**><!-- Opening tag College--> </**College**><!-- Closing tag College-->	

Figure 4. 12 XML with custom tags

Save the above document with your name **.xml (**such as **yourname.xml)** and keep updating the data and save your document after each step.

The next step is to organize the data of the college. Different colleges can have different structures or organizations for their data. Here we organize the college in terms of departments. The college can have several departments; in our case, it has three departments. We create again an element named <Department> as a sub-element of the <College>. The information may be organized as follows (See Figure 4.13).

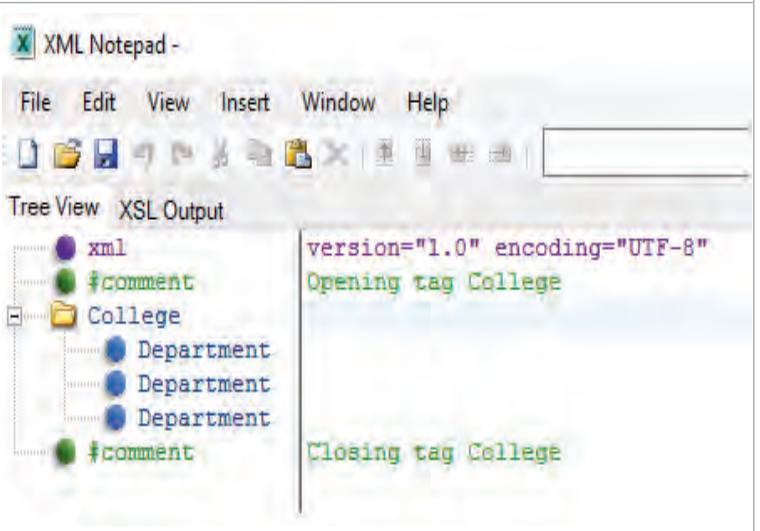
XML document Code	XML document view on XML Notepad
<?xml version =”1.0” encoding=”UTF-8” ?> <!-- Opening tag College--> <College> <Department> </Department> <Department> </Department> <Department> </Department> </College> <!-- Closing tag College-->	

Figure 4.13 College structure with Departments

The departments in the college can have detail information. This may include the department name and the department number (See Figure 4.14). Before proceeding to other sub-elements of the department, we can fill in the content for the department name and number. For further processing, you can attach data type to your elements. If you see the first department as an example, the attribute or data type for the department number is an integer value. Therefore, you have to think about your values.

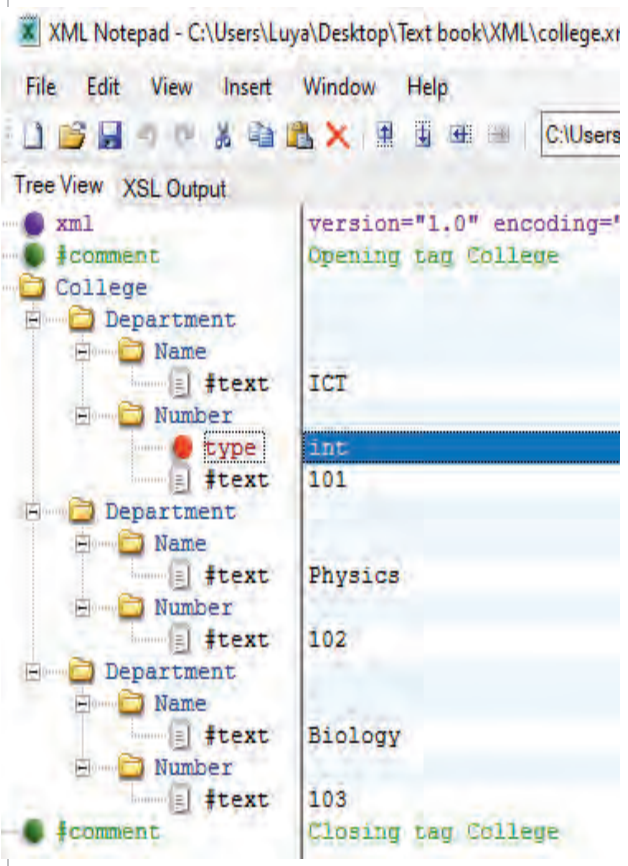
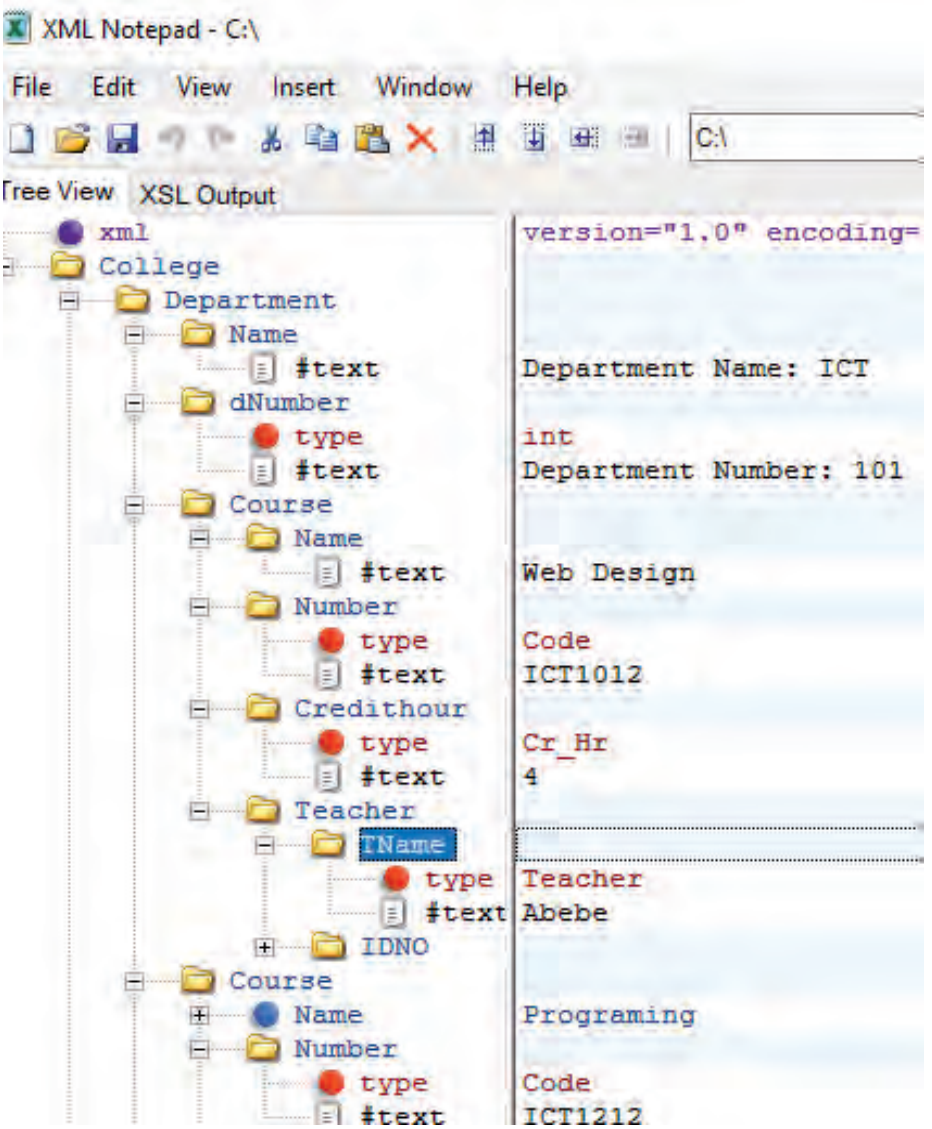
XML document Code	XML document view on XML Notepad
<?xml version =”1.0” encoding=”UTF-8” ?> <!-- Opening tag College--> <College> <Department> <Name> ICT </Name> **<Number type=”int”>101 </Number>** </Department> <Department> <Name>Physics</Name> <Number>102</Number> </Department> <Department> <Name>Biology</Name> <Number>103</Number> </Department> </College> <!-- Closing tag College-->	

Figure 4.14 College structure with Data Type Added

Not only the name and number of departments can be added but also details of students, teachers, and courses. Each of these can also be created as sub-elements of the department as shown below in Figure 4.15.

XML document view on XML Notepad



XML document

```
<?xml version =”1.0” encoding=”UTF-8” ?>
<College>
<Department>
        <Name>Department Name: ICT    </Name>
        <dNumber type=”int”>Department Number: 101 </dNumber>
        <Course>
                <Name> Web Design </Name>
                <Number type=”Code”>  ICT1012 </Number>
                <Credithour type=”Cr_Hr”> 4 </Credithour>
        <Teacher>
                <TName  type=”Teacher”>Abebe</TName>
        </Teacher>
        </Course>
        <Course>
        <Name>Programing</Name>
        <Number type=”Code”>  ICT1212 </Number>
        <Credithour type=”Cr_Hr”> 4 </Credithour>
        <Teacher>
        <TName type=”Teacher”>Ayele</TName>
        <IDNO type=”IDNO”> ICT/2323/13 </IDNO>
        </Teacher>
        </Course>
        <Student></Student>
</Department>
<Department>
        <Name> Department Name: chemistry </Name>
        <dNumber>Department Number:  102 </dNumber>
        <Course>
        <Name> Organic Chemistry </Name>
        <Number type=”Code”>  CHEM1212 </Number>
        <Credithour type=”Cr_Hr”> 4 </Credithour>
        </Course>
        <Course>
        <Name>Inorganic Chemistry</Name>
        <Number type=”Code”>  CHEM2121 </Number>
        <Credithour type=”Cr_Hr”> 4 </Credithour>
        </Course>
        <Teacher></Teacher>
        <Student></Student>
</Department>
</College>
```

Figure 4. 15 College Structure with Teachers’ and Students’ Data Added

Also, you can add teachers and students' data by creating sub-elements such as name, sex, age, etc.

Once you have designed your website, the next step is to publish it.

Until this point, you have learnt website design. Once you have completed the design of your website, the next step involves publishing the website so that the website is accessible worldwide by anyone.

Website publishing is the process of publishing the website's original content onto the Internet, or specifically onto a remote server. The term sometimes refers to the whole process of website design and publishing. This includes building and uploading websites, updating the webpages, and posting content to these webpages online. Web publishing includes personal, business, and community websites. The content meant for web publishing can include text, videos, digital images, artwork, and other forms of media such as music.
The most common thing about any website is that it is represented by a root directory. A root directory contains folders to organize images(.jpg,.gif), style files(.css), and script files(.js) that are used in the website, and its index file (See Figure 4.16 below).

1. **root**: this is the top level or root folder that represents the website itself. It contains all other files and folders of the website. For example, moe.gov.et.
2. **index.html**: contains the main homepages of the site which are written in HTML. Web servers are by default set up to return to the index.html file if no file name is specified. For example, if you write moe.gov.et, it returns to moe.gov.et/index.html.
3. **pages folder**: this subfolder contains web pages of the site. For example: about, contact us, etc. The name could be anything related to the site.
4. **images folder**: this is the subfolder that contains all the images that are used on the site.
5. **CSS folder**: CSS code used to style the site resides in this folder. It includes, for example setting text and background colors.

6. **scripts folder**: this subfolder contains all the JavaScript code used to add interactive functionality to the site (e.g. buttons that load data when clicked).

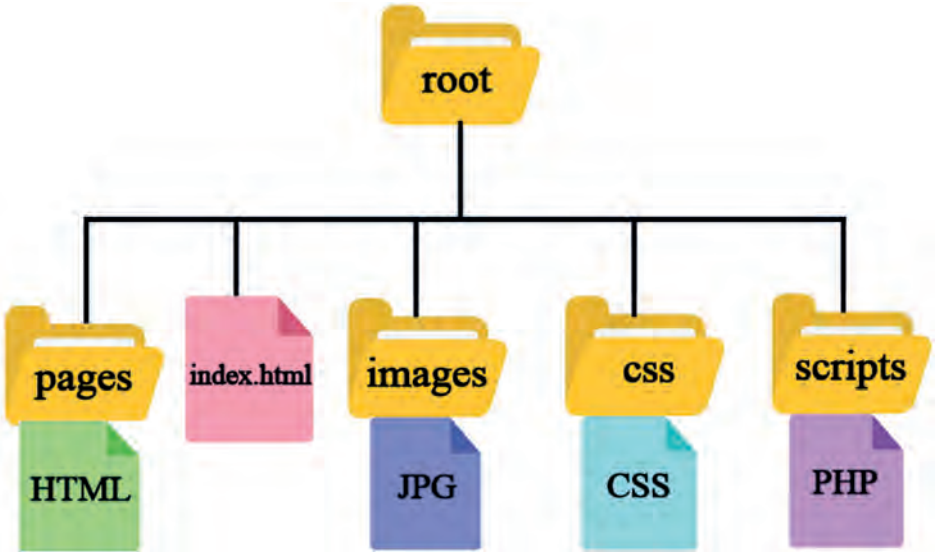


Figure 4. 16 Sample website - directory and files organization

A website is published by uploading website content or files onto the remote server which is provided by the hosting company or web host. Hosting companies provide web hosting services, which means providing online space for the storage of websites. A website is made available via World Wide Web (WWW). Web hosts must possess a web server. The web server is the actual location where your website resides. A web server may host single or multiple sites depending on what hosting service you have paid for.

The process of publishing a website also involves registering a domain name. A domain name is the part of your internet address that comes after "www". For example, in http://www.moe.gov.et/, the domain name is **moe.gov**. A domain name becomes your online business address, so care should be taken when selecting a domain name. Your domain name should be easy to remember and easy to type. It must be unique. If the one you want to use is taken or not available, domain registration fails, and you need to find another one.

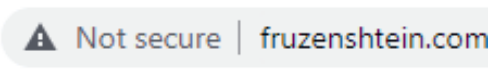
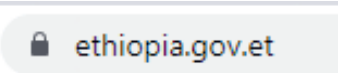
A domain extension is made up of three letter (for example, .gov in the above URL) at the end of the internet address which is known as a top-level domain names. The most common domain extensions include:

- .com for commercial sites.
- .edu for educational institution.
- .gov for government institutions.
- .org for a non-profit organization.
- .mil for military.
- .net for network

Security options

To keep your site safe and secure, a secure URL is needed. Particularly, if the site visitors are providing their private information, HTTPS is required, not HTTP. HTTPS (Hypertext Transfer Protocol Secure) is a protocol that is used to provide security over the Internet. To enable HTTPS, your website needs an SSL. SSL stands for Secure Sockets Layer which provides a secure online connection, and your website needs an SSL Certificate. SSL is also another necessary site protocol. It ensures your site visitor's personal information transfers between the website and your database are secure. SSL encrypts information (send from you/receive from the server) to prevent others from accessing and reading it while in transit.

To check whether a website is secure or not, you can type the website name into a browser address bar (e.g. ethiopia.gov.et) and notice one of the two results as shown below. A secure site displays a locked keypad (See the image in the second column below). If the site is not secure, before the web address, you could see the information icon and 'Not secure' (see the image in the first column below).

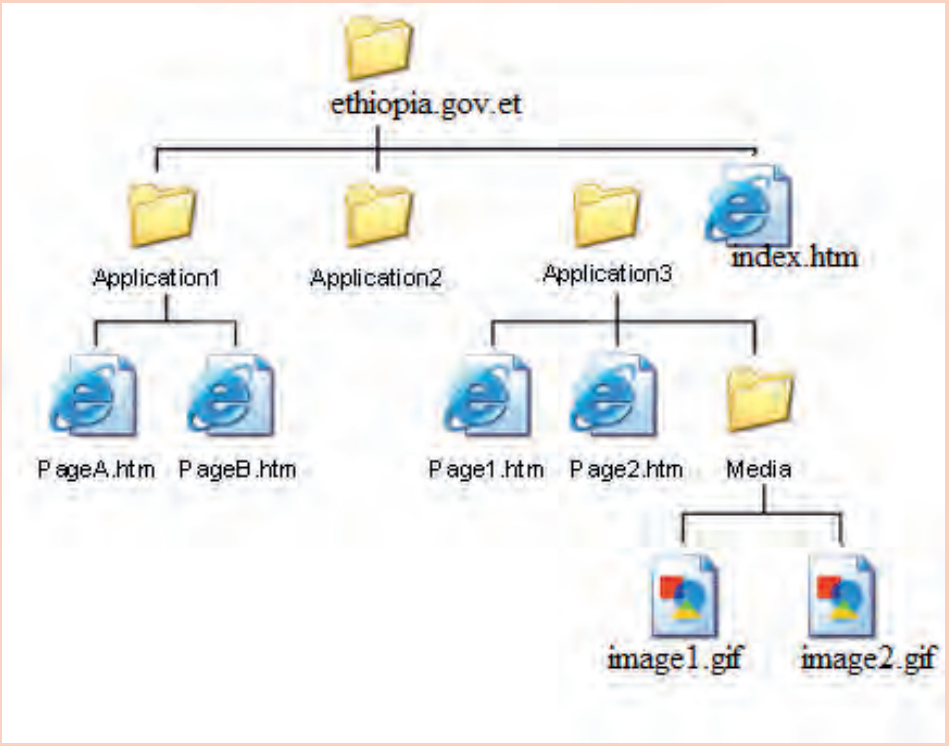
Non secure website is shown on a browser address bar	Secure website is shown on a browser address bar
	

Activity 4.9

1. Based on the below website directory:

   a. What is the name of the website?

   b. Identify the root directory and sub-directory, image directory, style directory, home page, or index page.



2. Take the websites of five Ethiopian federal institutions and check if their websites are secured. Hint: moa.gov.et, mofed.gov.et, etc.
3. Review at least two web hosting companies in Ethiopia, with features provided.

Unit Summary

In this unit, you have learnt about:

- the markup language XML.
- the definition of XML, its features, and purpose.
- elements of XML.
- XML vs. HTML – similarities and differences.
- advantages and disadvantages of XML.
- XML tags, root, XML declaration.
- document Type Declaration (DTD), Attributes, Comments.
- ways of opening XML file directly.
- entity references and entity declaration.

Key Terms

A **markup language** is a type of computer language that uses tags such as "<" and ">" to define elements in a document.

XML stands for **E**xtensible **M**arkup **L**anguage which is a way to apply structure to a web page. XML provides a standard open format and mechanisms for structuring a document so that it can be exchanged and manipulated.

An **XML document** contains the following: XML declaration, Document Type Declaration (DTD), Internal DTD subset, XML information set / Content, Root element, Tags, Data, Attributes, and Comments.

Points to remember while you work on XML:

- You should have one root in an XML document.
- XML elements are case-sensitive.
- XML should not overlap or all elements must properly be nested within each other.
- XML attribute values must always be in quotation marks.
- Comments in XML follow the HTML comment structure.
- XML does not truncate multiple whitespaces.
- Special characters need entity references, otherwise, they create errors.

HTML and XML are two different markup languages. HTML is the markup language to design and create web content, whereas XML is a markup language designed to store data. It is popularly used for the transfer of data. However, the two complement each other.

Entity References: This refers to characters that have special meaning in XML. There are 5 pre-defined entity references in XML, such as < (<), > (>), & (&), ' ('), and " (").

You can open and edit XML with any text editor, and view it with any web browser, or use a website that lets you view, edit, and even convert it to other formats.

In DTD the !DOCTYPE - defines the root element of the document.

PCDATA in<!ELEMENT releases (#PCDATA)>- means parsed character data.

Advantages of using **XML**: It makes documents transportable, separates data from HTML, flexible platform change process.

Disadvantages of using **XML**: It requires a processing application, syntax sometimes confusing, no intrinsic data type, and redundant syntax.

Website publishing is the process of publishing the website's original content on the Internet, or specifically on a remote server.

Websites are published by uploading website content/files onto the remote server which is provided by a hosting company or a web host.

SSL and **HTTPS** are protocols that provide security options that keep your site safe and secure. HTTPS is secured and prevents interceptions and interruptions from occurring while the content is in transit. The website requires an SSL certificate to enable HTTPS.

Review Questions

Part I: Match the items given under column B with associated items in column A

A	B
1. XML declaration	A. <body>
2. HTML tag	B. Extension of XML document
3. *.xml	C. Secured website
4. https://www.moa.gov.et	D. <DOCTYPE … […]>
5. XML attribute	E. <?xml version =”1.0” encoding=”UTF-8” ?>
	F. html
	G. <student id=’101’></student>
	H. <element/>

Part II: Choose the correct answer from the given alternatives.

1. Which of the following statement is true about XML?
   A. Elements are not case-sensitive.
   B. Quoting attribute is optional.
   C. Elements may nest but not overlap
   D. All of the above.

2. If you want to start XML coding, what is the correct declaration syntax for the version of an XML document?
   A. </xml version=”1.0”/>
   B. <?xml version=”1.0”/?>
   C. <xml version=”1.0”>
   D. <!Element />

3. What does DTD stand for?
   A. Dynamic Type Definition
   B. Document Type Definition
   C. Do The Dance.
   D. Direct Type Definition.

4. One of the following does not conform to element naming rule in XML documents.

   A. <Note>
   B. <h1>
   C. <1dollar>
   D. <NAME>

5. Among the available mark-up languages, XML is a complement to ___.

   A. HTML
   B. XHTML
   C. XQuery
   D. PDF

6. Which of the following is incorrect?

   A. The objective of XML is to replace HTML.
   B. Attribute of element should be in single or double quotation marks.
   C. Entity declaration is declared in DTD.
   D. All XML documents should begin with XML declaration.

7. Which of the following provides secured web access for a website?

   A. HTTP
   B. XML
   C. SSL certificate
   D. HTML

8. Which of the following XML documents is well-formed?

   A. <firstElement>some text <secondElement>another text </secondElement></firstElement>
   B. <firstElement>some text</firstElement><secondElement> another text</secondElement>
   C. <firstElement>some text<secondElement> another text</ firstElement></secondElement>
   D. </firstElement>some text</secondElement>another text<-firstElement>

9. An XML document is a well-formed XML document, when it___.

   A. contains a root element.
   B. contains an element.
   C. contains attribute.
   D. contain entity declaration.

10. Which one of the following is not the advantage of XML compared to HTML?

    A. Browser interfaces are simple to build.
    B. You can exchange data quickly between different platforms.
    C. You can use many tags to make a webpage.
    D. It is not object-oriented.

Part III: Code Writing

Take the environment where you are living and design the data structure for a specific organizational or environmental concept. For example, you may take animal kingdoms and structure their data, you may take Kebele residences and structure their data, or any other you feel appropriate.

Project: if you take, for instance Kebele residences, based on that:

- Try to find the structure of the documents in the kebele system
- Develop different pages/documents of the kebele

UNIT 5

MAINTENANCE AND TROUBLESHOOTING

Learning Outcomes

At the end of this unit, students will be able to:

- Practice installing and uninstalling software
- Explain software troubleshooting
- Describe network troubleshooting steps
- Identify network maintenance tools

Unit Overview

To ensure the safety and security of computer systems, every system user should take preventative actions including software preventive measures. Software preventive maintenance includes upgrading and installing software updates for security and reliability purposes. With a strong software preventative maintenance activity, time and money can be saved on one hand and failures can be reduced on the other. In this unit, you will learn about the installation and uninstallation of software, techniques used in software preventive maintenance, and basic maintenance tools that can be applied in our day-to- task routines .

5.1. Install and uninstall software

Notes

- In this unit, most hands-on activities are prepared based on windows based operating systems because of its user base and graphical based ease to work functionality to meet the required competency. But other operating systems can also function mentioned list of activities by following respective instructions for each operating system. It is also important to note that, there might be slight steps and functionality differences even within same operating system versions. Students are advised to refer the required steps and requirements based on types of operating system and current software versions installed on the computer devices.

Brainstorming 5.1

- Do you think the number and types of installed applications will have an effect on the speed and efficiency of your computing devices?

Users are able to perform some truly wonderful things with their computers and other digital devices. All of these activities including reading books, digital photo editing, using online social media sites, playing computer games, video streaming etc. are made possible by various kinds of software's. You can use computers or other handy held smart devices for even more activities by installing the required applications thanks to the new software apps that developers are constantly developing. Software can also be uninstalled when required following the proper steps and instructions.

5.1.1 Installing Software

Installation, is basically the process of getting the program ready to run including configuring hardware or software. It can also be referred as setup of computer software which includes device drivers and software plugins.

A device driver is a special kind of software program that controls a specific hardware device attached to a computer. Appropriate device drivers are essential

for devices to work properly on a computer with corresponding devices.

Steps for installing computer programs.

You can purchase a software program from software stores or galleries, download it from the source, it can be shipped to you. Software comes on CD, DVD, or USB flash disks. It is also common to download application software's for smart devices (Smart Phones, Smart Watches, Personal Digital Assistances, Smart TV and more) from software's stores like Apple (App) store, Google play store, Amazon App store, Samsung Galaxy Apps, Huawei App store etc. In all cases, please make sure to download or receive all software's from reliable and trusted sources only to protect your machines from malicious software's.

Links

Refer Unit 5 of the Grade 10 IT Textbook Information and Computer Security to revise concepts related to protecting computing machines from malicious software's.

Scenario 1: Installing software's on personal computers

Step 1: Insert the installation CD /DVD into the PC's optical drive or insert the USB drive on the USB port.

Step 2: Run the installation program.

The installation program runs automatically when you insert the disc (See Figure 5.1), or, you see the AutoPlay dialog box, from which you can choose the Install or Setup command. If the installation program does not run automatically, follow these sub-steps :

- Open the Computer window
- Right-click the drive's icon.
- Choose from the menu the command Install or Run Program from your media. If that command is not available, choose the AutoPlay command.

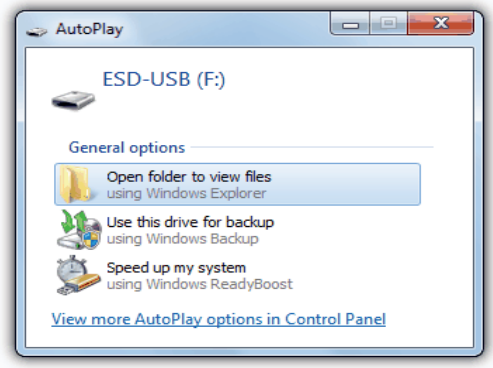


Figure 5.1. AutoPlay Box

Step 3: Follow the instructions on the screen.

Read the information carefully; sometimes they slip something important in there.

You may also be hit with a User Account Control (UAC) warning here. If so, type the administrator password or click the Yes or Continue button to proceed.

Users have to agree to abide by the software license. The software may ask for a serial number. It can be found somewhere inside the box, in the manual, or on a piece of paper or on shipment email if it is purchased from online stores. If you are asked to, replace one disc with another. This process may go on for some time. You may also need to identify and locate the place to install the files in.

Step 4: Wait for the Installation program to end. You see a message telling you whether the installation is successful or not.

Step 5: Close the installation program window.

Step 6: If prompted, restart Windows.

Step 7: Start using the program

Notes

- The above steps are for Windows operating systems; other operating systems including Unix/ Linux, Mac etc. have their procedure that you need to follow while installing software.

Scenario 2: Installing software's on Mobile Devices

Worldwide, mobile devices has grown significantly, creating new paths for communication for individuals. Its acceptance has allowed increased access to public information and fundamental services to people irrespective of time and location powered by numerous apps and gadgets. This requires the right skill and attitude to select, install and use application software's.

It is so important to find the right source before downloading any apps on our mobile devices (Android, Apple or other devices). You can download no-charge and paid apps from application stores based on your device operating systems. You can also download apps form other unknown sources on the internet when necessary. Please use an internet and browse the appropriate app stores to download apps to mobile devices (android / non android based mobile devices) and other smart devices.

Notes

- Your phone and personal information may be subject to damage if you download apps from unknown sources. This includes the possibility of data loss or damage to your phone, as well as the obvious possibility of damage or hacking to your personal information. Review your phone's security settings, which scan for potentially harmful apps, alert you when one is found, and delete the program if necessary.

Activity 5.1

1. What are major issues that should be checked before installing software's on our computers or other mobile devices?
2. Based on your previous experiences with installing applications, discuss which steps were different, and reasons for differences if any.
3. Select one application software that can be used to improve your digital literacy skills and install it to available device (computers or mobile devices). Share the procedures to get source of the software and followed installation steps during installation.

5.1.2 Uninstalling Software

Uninstalling unused or unwanted programs can help your computer in various ways. It can optimize hard drive spaces and increases computer speed.

Software installed on computers or mobile devices must be properly uninstalled and users should know the reasons to un install the software program before taking actions. This becomes so important when it applies on computer and other devices which belongs organizations, schools, libraries and computer laboratories where different types of users appears with their own software requirements. This also applies when you use other individuals' personal devices including your friends and families. It is recommended to consult computer technicians or owner of the devices before making any software uninstallation process .

Scenario 1: Uninstalling software's on personal computers

Here is how it works:

- Open the Control Panel.
- Click the link Uninstall a Program, found beneath the Programs heading.
- The Programs and Features window appears. It lists all software installed on your PC.
- Select the program you want to uninstall.
- Click the Uninstall/Change button on the toolbar.
- If prompted by a User Account Control, type the administrator password or click the Continue button.
- Continue reading the instructions on the screen to uninstall the program.

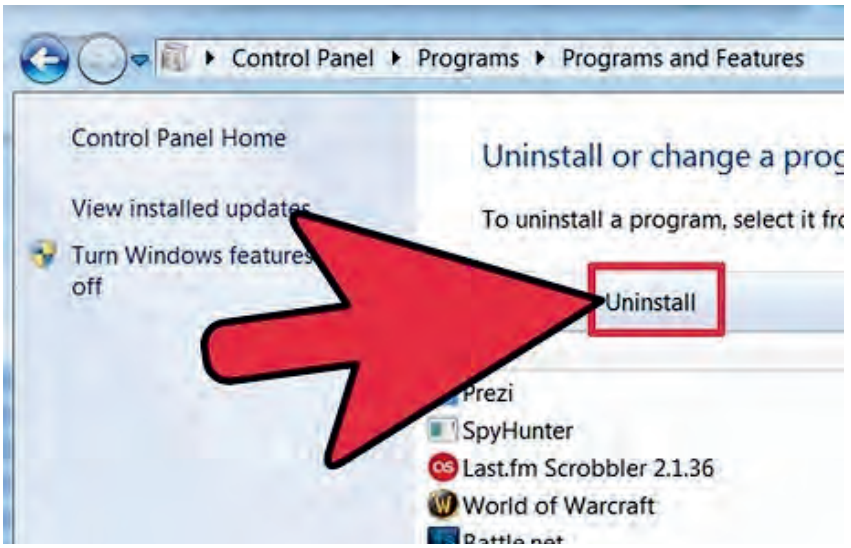


Figure 5.2. Software Uninstall process on Windows

The uninstall directions vary from program to program, but eventually, the program is removed.

Scenario 2: Uninstalling software's on Mobile Devices

You can uninstall or delete apps you have installed on your mobile phone. You can also disable system apps that came with your phone.

There are few steps to uninstall apps or delete apps on our mobile devices and it is some what similar for most phones operate with different mobile operating systems including Android, Window, Samsung, Apple, Blackberry and more. Commonly, you can easily uninstall from manage apps and devices setting of the device, taping the name of the apps to delete and then, uninstall. Most users also choose to swipe up on the screen to access the apps, then long press the app icon for more options and select Uninstall.

Activity 5.2

1. Discusses the moments and reasons you plan to uninstall computer applications based on your previous experience.
    a. What are the major reasons for uninstalling the apps?
    b. List steps you followed and is there any difference from what is stated above? What do you think are the reasons for these differences?
2. Consult your IT teachers and make a review of installed applications in your computer laboratory and identify potential applications that could to be uninstalled to optimize the computer performances. Present your lists and recommendations for the class.
3. Have you ever tried to uninstall some applications on your mobile devices but you can't make it? Share your experiences among your classmates. Why do you think you can't uninstall some apps on mobile devices?

5.2. Software Troubleshooting

Brainstorming 5.2

- What would you do if you opened your computer and discovered that you lost all of your files?

Troubleshooting is a systematic approach to locating the fault and identifying the cause of the fault in a computer system.

In the case of computer systems, the term troubleshoot covers both hardware and software issues. Hardware troubleshooting includes identifying and fixing the problems when the hardware devices like mice, monitors, printers, modems, etc. are malfunction. Software troubleshooting includes identifying and fixing problems encountered while working with the operating system and application softwares. It also includes protecting our computer systems from malicious damages including viruses and worms.

Figure 5.3: Hardware Troubleshooting

Not all troubleshooting processes are the same, and all users tend to refine their troubleshooting skills based on knowledge and personal experience to manage some common issues that can be fixed even in the absence of professional technicians. Troubleshooting is a learned skill.

Figure 5.4: Software Troubleshooting

Activity 5.2

1. Assess the most common software problems caused by viruses and worms and present your findings to the class.

2. What can you do if your computer become completely unresponsive, or frozen and you will not be able to click anywhere on the screen, open or close applications, or access shut-down options ? Discuss the possible causes for the problem and possible solutions with your pair and present your findings to class.

Even though there are constantly new software and IT issues emerging, the better you understand these issues, the simpler it will be for you to fix them. Office systems can be slowed down by corrupted software and incorrect software, which can cause delays, general inefficiencies, and a lack of productivity. Let us see few common software problems with their possible troubleshooting techniques.

Example 1 - An application is running slowly or application frozen

Possible Solutions: *The first step to try is to close and reopen the application. If the problem persists*, update the application. To do this, click the Help menu and look for an option to check for Updates. If you don't find this option, another idea is to run an online search for application updates (See Figure 5.5).

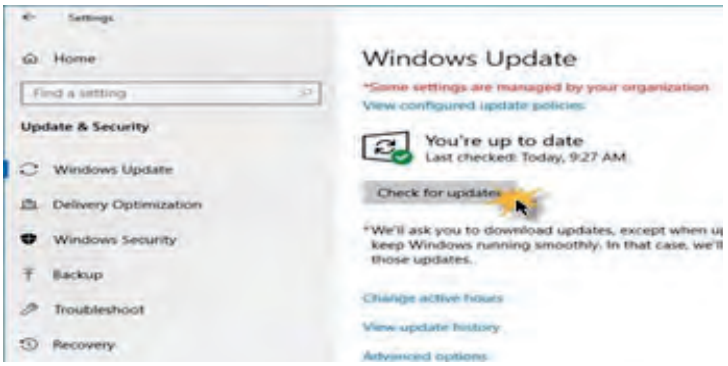


Figure 5.5 Window Update

Similarly, sometimes an application may become stuck, or frozen. When this happens, you will not be able to close the window or click any buttons within the application. This can be solved with force quit the applications. On a computer, you can press (and hold) Ctrl+Alt+Delete (the Control, Alt, and Delete keys) on

your keyboard to open the Task Manager (See Figure 5.6) under the Processes tab, click the frozen application and click the End Task button. If you are unable to force quit an application, restarting your computer will close all open apps.

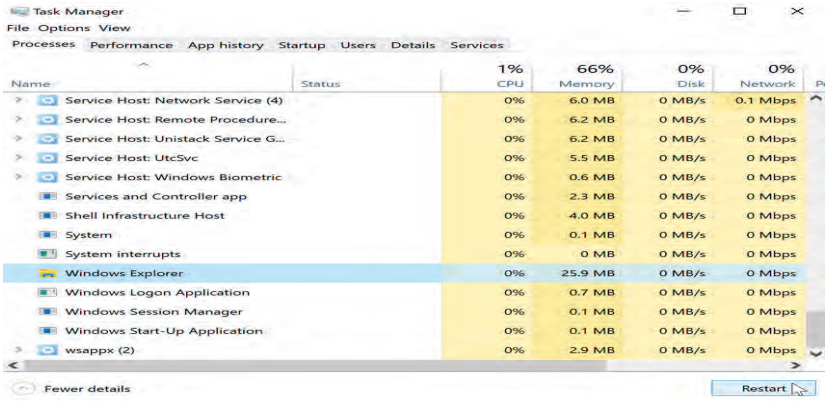


Figure 5.6 Task Manager

Example 2 - All programs on the computer run slowly or computer is frozen

Possible Solutions: You may have malware running in the background that is slowing things down which needs a virus scanner with anti-virus or anti malware software's. Beside this, your computer may be running out of hard drive space which needs deleting any files or programs you do not need. If you are using a PC, you can run Disk Defragmenter and see the available disk spaces and fragment the disk space. You will learn how to disk defragment later in this unit.

Example 3 - The sound is not working

Possible Solutions: First, check the volume level. Make sure the sound is on and the volume is up, then click the audio button. It's also a good idea to check the controls on the audio player. There are many players that include separate audio controls for audio and video. Ensure that the player's volume is set to the highest setting and that the sound is switched on. Check that external speakers are plugged in, switched on, and connected to the appropriate audio port or a USB port if the issue persists (See Figure 5.7). Make sure the wireless headphones are successfully paired and connected in this scenario. Sound drivers' difficulties can also be related to the other prevalent causes. The driver version you have should be shown in device management. Next, carry out a manual update push. When you've found the audio driver, right-click it and choose Update Driver if it hasn't already been

updated; if not, acquire the most recent sound driver for the current operating system version from reputable sources.

Figure 5.7: Headphone plugged into laptop

Activity 5.3

- Identify possible solutions for computers with black screen
- What are the possible solutions for this problem?

Tip

If you have not been able to solve your challenge, you may need to ask for assistance from others. Using the Internet is recommended as a starting point. It is likely that other people have likely encountered similar issues, and solutions are frequently shared online. Also, if you have a friend or family member who is knowledgeable about computers, he or she may assist you, so you do not afraid to ask.

Generally, any issues can arise anytime and therefore you should know how to avoid any problem from happening as part of preventive maintenance . But when happened, knowing where to start troubleshooting can save you lots of time. Most problems are software problems. Some are definitely hardware problems. And some can be caused by one or the other. Sometimes, our software problem can be easily resolved by running a repair utility or uninstalling and reinstalling. It is always appreciated to know the proper troubleshooting skills to save out time and resources. Internet can be a good resource to see similar problems and proposed solutions for most problems easily to start with. This will also help you

to acquire better digital literacy skill which is one of the fundamental 21st skill set for all professionals at all disciplines. Other problems may need professional experts support to fix and maintain problems, so it is recommended to consult those professionals and technicians based on the level of the problem and your ability to fix it by your own.

5.3. Tools to speedup computer system

Brainstorming 5.3

- How does installing window updates in regularly help to maintain the computer health and functionality?

The system tools for computer maintenance are:

- Disk defragmenter tool
- Disk cleanup tool
- Windows update

5.3.1 Disk Defragmenter

Over time, your disk may be fragmented. By fragmented disk, we mean files are broken and stored in unmanaged space of the disk and hence the computer becomes slow to bring the fragments together in order to open the files.

Disk Defragmenter is a system tool used to rearrange the files on the disk. Defragmentation reduces file system fragmentation and, increases the efficiency of data retrieval and thereby improves the overall performance of the computer. It rearranges all the files so that each file is stored in one sequence (See Figure 5.8).

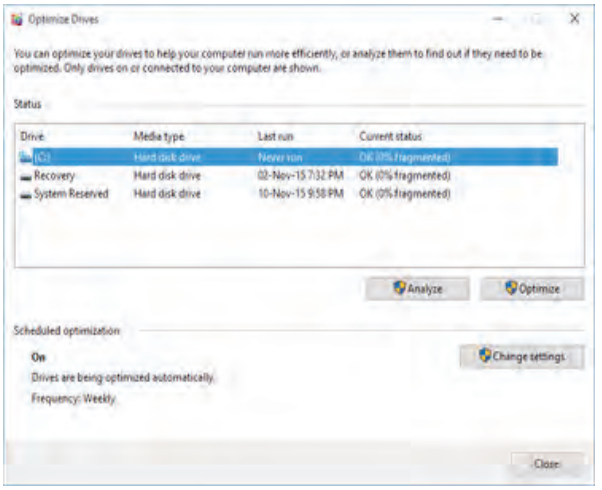


Figure 5.8. Disk Defragmenter

Notes

- Type '**Disk Defragment'** on your window search box to access it.

5.3.2 Disk Cleanup

Disk Cleanup system tool deletes all the unused and unwanted files on the hard disk to free some disk space. The hard disk contains many files that are unnecessary after a certain amount of time, but they still reside on the disk. These need to be removed manually, or an automatic Disk Cleanup program can be scheduled (See Figure 5.9). Disk Cleanup deletes all of your unused and unwanted files. It also deletes your garbage and old Internet files. It should be performed weekly.

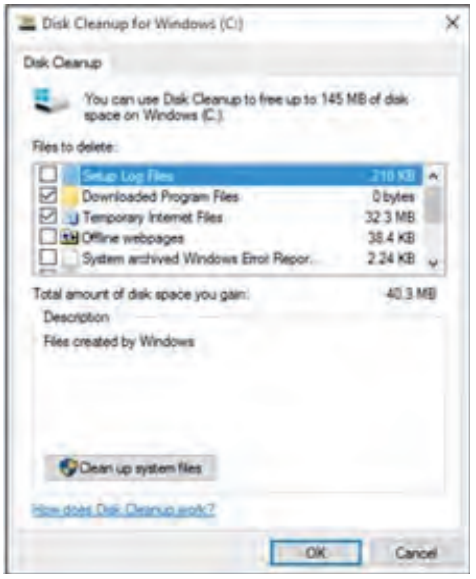


Figure5.9. Disk Cleanup

Notes

- Type '**Disk Cleanup'** on your window search box to access it.

5.3.3 Windows Update

Windows update ensures that your computer is up-to-date with new features and the latest security. This feature needs a connection to the Internet. It connects to the Microsoft server and downloads and installs the updates on the system. You can click on the Windows Update icon in the Start menu and click on the Download update to get the updates (See Figure 5.10).

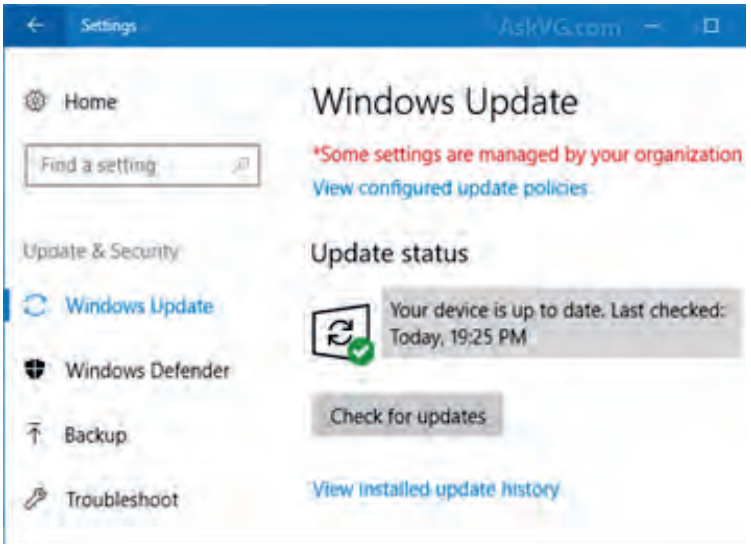


Figure 5.10. Window Update

Notes

- Type '**Window Update'** on your window search box to access it.

There are also additional techniques to maintain the computer

Check disk

Check disk fixes the errors on your computer when you are having problems or receiving errors. This should be run every week just to ensure that your computer is error free even if you are not having any problems. You can start the scan disk wizard, by clicking on the C drive, selecting properties from the menu, and then selecting the Check button (See Figure 5.11).

Figure 5.11. Disk Checkup

Notes

- Type '**Disk Checkup'** on your window search box to access it.

Close Unwanted Programs

Sometimes there are programs that you do not want to run anymore or programs that get halted due to some problems. Although the best way to close a program is to choose the 'Close' or 'Exit' command from the 'File' menu but there is another way of closing it.

Pressing the **'Ctrl+Alt+Delete'(See Figure 5.13)**keys simultaneously brings up the Close Program dialog box as illustrated below. You can close the program you want to close from it, by choosing it and then pressing the 'End Task' button as it can be seen in Figure 5.12. The program is closed at once.

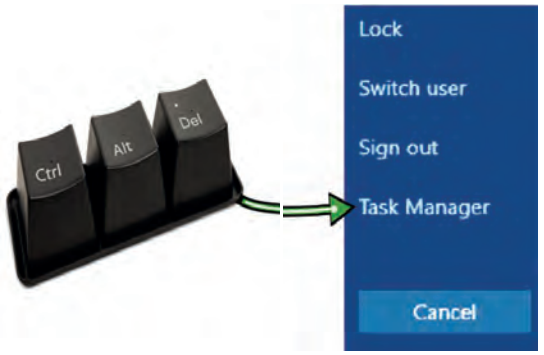


Figure 5.13: Opening Task Manager

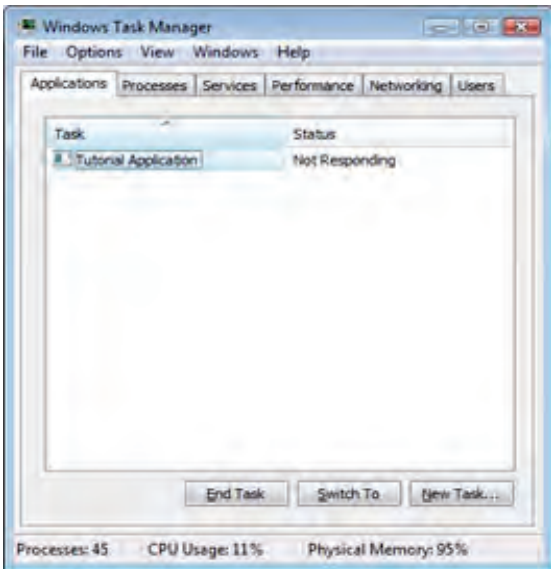


Figure 5.12. Window Task Manager

Activity 5.4

- Practice the windows operating system's built-in system tools including disk defragmentation, disk cleaning up, and cleaning unwanted programs in your computer laboratory activities.

5.3.4 System Restore

System restore is a system tool that is used to create a new restore point or to restore the system to a previous restore point. It makes a secure copy of the system settings when it is functioning normally, which is useful in case the system malfunctions.

Whenever you install a new software, it is always advisable to save the setting of the system so that at a later stage if the system malfunctions, then that saved copy of the system, can be restored. It helps in getting the system up and running for use. To make a restore point, choose the System Restore from the Start->Programs->Accessories->System Tools->System Restore. To restore the system to a previous, restore point and, follow the same steps, but just choose to restore rather than make a new restore point.

Common problems and solutions

Table 5.1. Common Computer problems and solutions

Problem	Solution
Slow Computer	Disk Cleanup Disk Defragmentation Close programs Possible risk of virus run an Antivirus program
Frozen screen	Close programs System restores Restart your computer Check disk

Activity 5.5

- Mention major system tools that are used to speed up computer systems and explain their benefits.

5.4. Network Troubleshooting

Brainstorming 5.4

- What are the possible causes for Internet network disconnection?

Two or more computers connected together to share resources (such printers and scanners), exchange files, or enable electronic communications make up a network. A network's connections to its computers can be made by cables, phone lines, radio waves, satellites, or infrared laser beams.

The collection of techniques and procedures used to locate, analyze, and fix issues in a computer network is known as network troubleshooting. Network engineers and technicians use a logical procedure to fix issues with networks and enhance network performance.

When you run a network or while working in any system, there are always chances of failure in the smooth operation owing to technical, physical, or any other faults. For the uninterrupted running of the system, you need to resolve these raised issues as soon as possible. Therefore, you need to detect the cause of the problem first and then fix it. The process of detection, minimization, and resolving the faults that arise in the network while performing the various day-to-day activities is known as network troubleshooting.

Because of the variety of network hardware, network configurations, operating systems, and setups, not all the following information may not apply to your network or operating system, but it is recommended to check the following common issues in network troubleshooting

Notes

- Network problems happen when something disrupts the connection between your computer and the content you are trying to access.

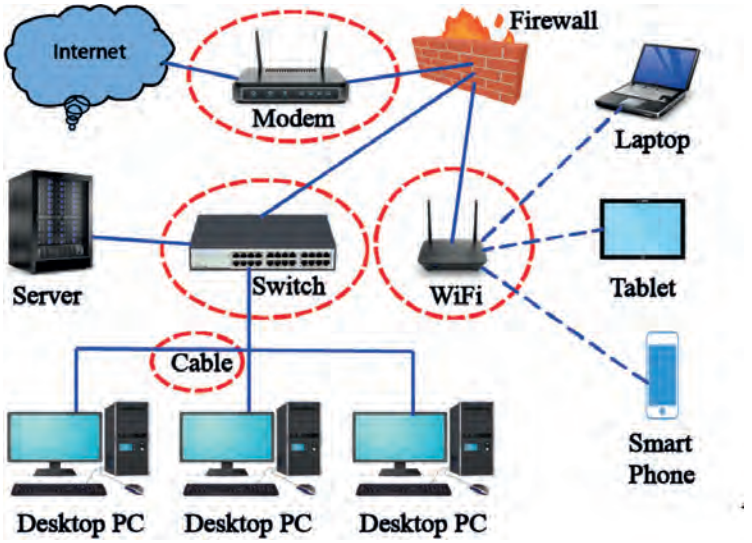


Figure 5.14: Common Network problems cable problem, switch failure, modem failure and Wi-Fi router failure

5.4.1 Basic Network Problems

Network problems include slow transmission of data and no connection at all. The problem can be caused by the following:

Cable Problem: The cable which is used to connect two devices can get faulty, shortened, or can be physically damaged (See Figure 5.15).

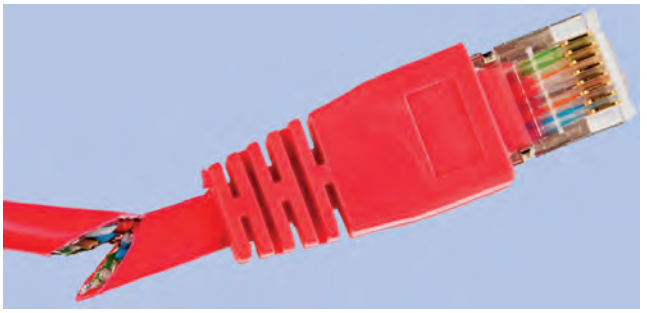


Figure 5.15: Cable problem

Connectivity Problem: The port or interface on which the device is connected or configured can be physically down or faulty due to this the source host can not communicate with the destination host.

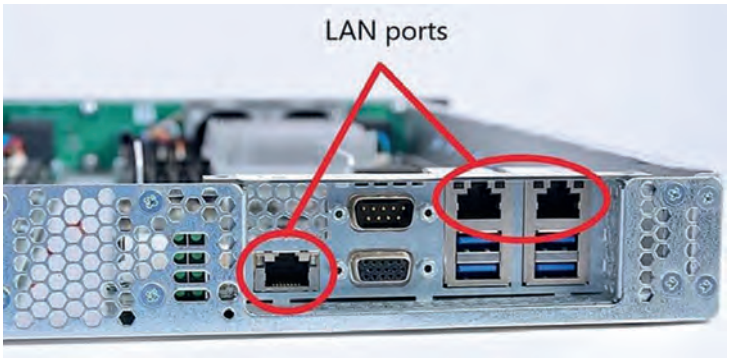


Figure 5.16: LAN ports

Out of Reach Issue: with wireless networks, the wireless host may be too far from the access point, or there could be an obstructing object between the access point and the wireless host.

Configuration Issue: Due to a wrong configuration such as; routing problem, and other configuration issues, network fault may arise and the services can be affected.

Software Issue: Owing to software compatibility issues and version mismatch, the transmission of IP data packets between the source and destination is interrupted.

Traffic overload: If the link is over-utilized, then the capacity or traffic on a device is more than its carrying capacity, and due to overload conditions, the device can start behaving abnormally.

5.4.2 Basic Network Troubleshooting Steps

The first step of solving a network problem is finding the source of the problem. Network problems are caused by a part of the network you control or outside your control. We assume that you control your local network and do not control anything beyond that. The way you solve the problem depends on whether or not you control the failing part of the network. You can solve the local network problems by yourself because you control the network. Outside network problems require help from whoever runs that network.

Activity 5.6

Write the meaning of local network and outside network

- Give an example of a local network and an outside network based on your context.
- Who manages the outside network in your geographical location?

Network troubleshooting is a repeatable process, which means that you can break it down into clear steps that anyone can follow (See Figure 5.17).

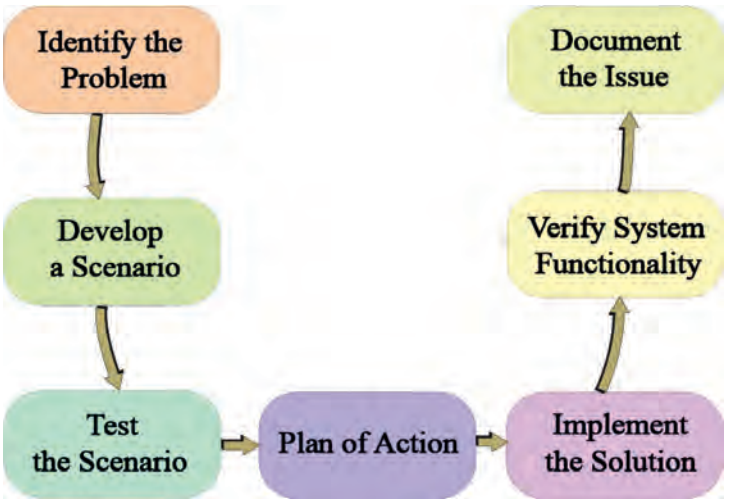


Figure 5.17: Basic Network Troubleshooting steps

- **Identify the Problem**

As discussed earlier, the first step in troubleshooting a network is to identify the problem. As a part of this step, you should do the following.

- **Gather information** about the current state of the network using the network troubleshooting tools that are available to you.
- **Duplicate the problem** on a test piece of hardware or software, if possible. This can help you to confirm where your problem lies.
- **Ask users** on the network to learn about the errors or difficulties they have encountered.

- **Identify the symptoms** of the network outage. For example, do they include complete loss of network connection? Does it slow behavior on the network? Is there a network-wide problem, or are the issues only being experienced by one user?
- **Determine if anything has changed** in the network before the issues appeared. Is there a new piece of hardware that is in use? Has the network taken from new users? Has there been a software update or change somewhere in the network?
- **Define individual problems clearly**. Sometimes a network can have multiple problems. This is the time to identify each issue so that your solutions to one is not bogged down by other unsolved problems.

- **Develop a Scenario**

Once you have finished gathering all the information that you can about the network issue or issues, it is time to develop a working scenario. Sometimes a network outage occurs because someone tripped on a wire or some other simple problem.

- **Test the Scenario**

Using the tools at your disposal, test your scenario. If your scenario is that the network router is defective, try replacing it with another router to see if that fixes the issue.

Notes

- *At this stage, it is important to remember that proving your own theories wrong does not mean that you have failed. Instead, you return to step two, develop a new scenario, and then find a way to test that one.*

- **Plan of Action**

Come up with a plan of action to address the problem. Sometimes your plan can include just one step. For example, restart the router. In other cases, your plan can be more complex and take longer, For instance, when you need to order a new part or roll a piece of software back to a previous version on multiple users' computers, you need more steps.

- **Implement the Solution**

Now you have a plan for fixing the network, it's time to implement it. There are some solutions that you may do by yourself, while others may require cooperation from other network experts or users.

- **Verify System Functionality**

Make sure that the issue in question has been resolved, but there is also on the lookout for other issues that may have arisen from the changes that you have made to the network.

- **Document the Issue**

Make sure to document each stage of troubleshooting the problem, including the symptoms that appeared on the network, the scenario you developed, your strategy for testing the scenario, and the solution you have come up with to solve the issue. Even if you do not reference this documentation, it may be helpful to other network technicians, students, or users at your school, home, or organization in the future and could help to shorten network downtime.

5.4.3 Network Troubleshooting Tools

When it comes to identifying and resolving network issues, you can utilize a variety of methods. These tools may be built into the operating system of the computer, available as stand-alone software programs, or available as hardware devices that you can use to troubleshoot network issues. In this sub-unit, tools that exist in your operating systems and network maintenance hardware tools are discussed.

5.4.3.1 Command-Line Tools

On Windows PCs, the command prompt can be accessed by searching for it in the start menu or by typing **"cmd" without the quotation marks into the Run Window (See Figure 5.18)**.

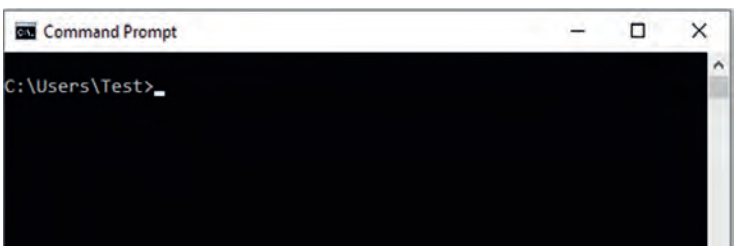


Figure 5.18: Windows Command Prompt

The following commands can be entered into the command prompt one at a time to reveal specific information about the network status:

ipconfig — A Windows Transmission Control Protocol /Internet Protocol (TCP/IP) utility that verifies network settings and connections. It can tell you a host's IP address, subnet mask, and default gateway, alongside other important network information.

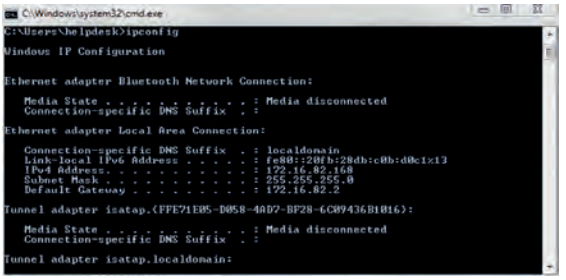


Figure 5.19: ipconfig command

ping — A TCP/IP utility that transmits a datagram to another host, specified in the command. If the network is functioning properly, the receiving host returns the datagram. See the fig 5.20 which shows ping the default gateway which is 192.168.0.1 in this case. Check your own default gateway and apply ping command to see the results.

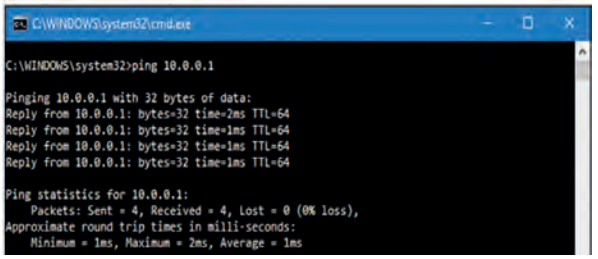


Figure 5.20: ping command

tracert —A TCP/IP utility that determines the route data takes to get to a particular destination. This tool can help you to determine where you are losing packets in the network, or helping to identify problems.

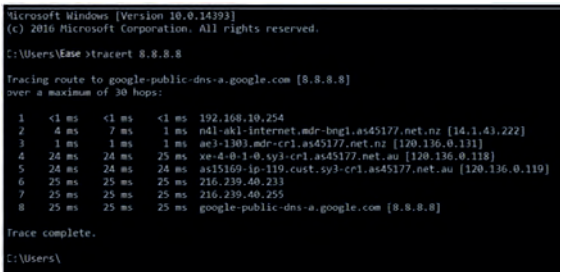


Figure 5.21: tracert command

nslookup — A domain Name System (DNS) utility that displays the Internet Protocol (IP) address of a hostname or vice versa. This tool is useful for identifying problems involving DNS name resolution. See Fig 5.22 which shows google dns ip address (8.8.8.8) with nslookup command.

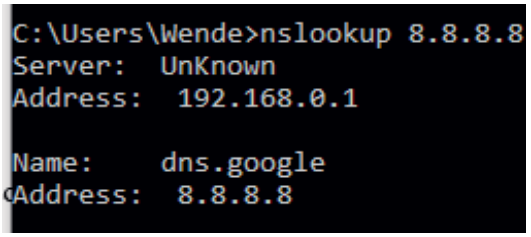


Figure 5.22: nslookup command

netstat — A utility that shows the status of each active network connection. This tool is useful for finding out what services are running on a particular system.

Notes

- In practice, there are many command line tools used for network troubleshooting for different operating systems including the windows operating system. You can search online to get, learn and apply different commands. Those mentioned commands for windows operating system, are subject to change when you need to apply to another operating systems like that of Linux/ Unix or Macintosh. Please browse the equivalent commands based on device own operating system if any. .

Activity 5.7

Practice the most common network troubleshooting command tools including; 'ipconfig', 'ping', 'tracert', 'netstat' and 'nslook' up in the windows operating system, but do not be limited to these tools.

- Record your results
- You can use Google's primary DNS server which is 8.8.8.8 to practice ping command or you can practice the command using your school Internet default gateway if applicable.

Notes

- You can get your device default gateway from your ipconfig command practice using cmd. Please also note that the IP address mentioned above can be changed due to server address changes by their respective service providers.

In addition to command-line tools, several standalone applications can be used to determine the status of a network and troubleshoot issues. Some of these applications include packet sniffer, port scanner, protocol analyzer, Wi-Fi analyzer, and more.

5.5. Network Maintenance Tools

Command-line tools and applications are software tools for troubleshooting, but some network problems have hardware causes and solutions. Here are some hardware tools that can help you diagnose and fix network issues:

Wire Crimpers - A wire crimper (sometimes called a cable crimper) is a tool that attaches media connectors to the ends of cables. You can use it to make or modify network cables (See Figure 5.23).

Figure 5.23: Wire Crimper

Cable Testers — A cable tester (sometimes called a line tester) is a tool that verifies if a signal is transmitted by a given cable. You can use one to find out whether the cables in your network are functioning properly when diagnosing connectivity issues (See Figure 5.24).

Figure 5.24: UTP cable tester

Punch Down Tool — A punch-down tool is used in a wiring closet to connect cable wires directly to a patch panel or punch-down block. This tool makes it easier to connect wires than it would be to do it by hand (See Figure 5.25).

Figure 5.25: Cable Puncher

Tone Generator — A tone generator is a device that sends an electrical signal through one pair of UTP wires. You can use these tools to verify that signals are passing through the wires in your network. They are often used to confirm phone connectivity (See Figure 5.26).

Figure 5.26: Tone Generator

Multimeter —is an electronic measuring instrument that takes electrical measurements such as voltage, current, and resistance. There are hand-held multimeter for fieldwork as well as bench-top models for in-house troubleshooting (See Figure 5.27).

Figure 5.27: Multimeter

Unit Summary

In this unit, you have learnt about:

- common steps of installing and uninstalling software
- basic preventive maintenance and fundamentals of software troubleshooting concepts
- common software-related problems and their alternative solutions
- basic tools to speed up computer system
- issues in system restore
- network troubleshooting and basic troubleshooting steps and tools
- common network troubleshooting applications and hardware maintenance tools

Key Terms

Disk defragmentation - is the process of reorganizing the data stored on the hard drive so that related pieces of data are put back together, all lined up in a continuous fashion.

Disk Cleanup - Disk Cleanup helps free up space on your hard disk, creating improved system performance. Disk Cleanup searches your disk and then shows you temporary files, Internet cache files, and unnecessary program files that you can safely delete.

Windows Update- keeps your computer up to date with security and bug fixes automatically, with little effort on your end. This protects your computer from malicious software and fixes known Windows bugs and other issues.

Installing software - is the act of making the program ready for execution.

Uninstalling Software – is a process of removing a program from your system which you previously installed in your system.

System Restore - is a feature in Microsoft Windows that allows the user to revert their computer's state (including system files, installed applications, Windows Registry, and system settings) to that of a previous point in time, which can be used to recover from system malfunctions or other problems.

Troubleshooting - is a systematic approach to locating the fault part and identifying the cause of the fault in a computer system.

Network troubleshooting – is the act of discovering and correcting problems with connectivity, performance, security, and other aspects of networks.

Review Questions

Part I: Choose the correct answer from the given alternatives.

1. The process of detection, minimization, and resolving the faults that arise in the network while performing the various day-to-day activities is _____.

   A. Network troubleshooting
   B. Computer maintenance
   C. Backup
   D. System restores

2. Basic network problems might include _____.

   A. Cable problem
   B. Connectivity problem
   C. Configuration issue
   D. All

3. Which one of the following is the first step to solving a network problem?

   A. Finding the source
   B. replacing the faulty part
   C. Refer the manual
   D. Resolve the issue

4. In network troubleshooting, once you have finished gathering all the information that you can about the network issue or issues. What is the next step to do?

   A. Develop a scenario
   B. Test the scenario
   C. Document the issue
   D. Verify system functionality

5. Which command line tool is used to get a host IP address, subnet mask, and default gateway?

   A. tracert
   B. ipconfig
   C. netstat
   D. ping

Part II: Give short answers to the following questions

1. What is preventive maintenance, and why do we need it?
2. What are the possible sources to get application software?
3. What sort of precaution shall be considered before download application software from Internet sources?
4. Mention and explain common command line tools that can be used for troubleshooting
5. Write the purpose of cable crimpers and cable testers.

Part III. Case Study

1. Assess the most common software problems you face in your computer laboratory in a group and prepare a summary of common software and network-related problems and your counter-solution alternatives.

Remark *- Keep in mind that most computer and network problems have simple solutions, although it may take some time to find them. For difficult problems, a more robust solution may be required including reformatting your hard drive or reinstalling your operating system. If you think you might need a solution like this, we recommend consulting your IT teacher and IT technician(s) for support and guidance.*

UNIT 6

FUNDAMENTALS OF PROGRAMMING

Learning Outcomes

At the end of this unit, students will be able to:

- Explain program flow controls
- Describe conditionals program flow control
- Describe iteration program flow control
- Construct program statements using control statements
- Appreciate Python interpreter using Integrated Development Environment(IDE)
- Analyze program statements in debugging mode

Unit Overview

A software program is written using a programming language. More than a hundred programming languages have been introduced since the introduction of the computer. Programming languages are designed with a specific purpose. Like human languages, programming languages have their own specific set of rules called syntax, which define how to write instructions in that language. Python relies on indentation to define scope and uses new lines to complete a command. The semantics of a Python program refers to the meaning of the set of rules. In Grade 11, you were introduced to programming language and its generation, basic syntax in python, and program writing.

In this unit, additional language constructs and concepts that can help you develop a program in Python are introduced. First, Python for program flow controls, conditionals, and loop statements are discussed followed by comments, which enable you to provide some text description about your program. Finally, interpreter and debugging the Python program are discussed.

6.1. Program Flow Controls and Syntax in Python

Brainstorming 6.1

- In what order do instructions in a Python program execute?

A program flow control is the order in which the program's code executes. In Python, the flow control of the program is regulated by the implementation of three types of programming language constructs or program logic, namely *sequential*, *branching*, and *loop*. The sequential flow control consists of a simple list of statements that are to be executed in the order given. Everything you learnt about declaring variables, printing output, and reading input from the keyboard in Grade 11 are sequential statements. In this unit, the implementation of the remaining two language constructs: conditional and loop flow controls are focused.

Activity 6.1

- Discuss the three types of programming language constructs and connect them with real-world problems.

6.1.1 Conditionals Program Flow Controls

Brainstorming 6.2

- "If it rains, then I will take an umbrella." – what type of statement is this?

In some cases, you may need a program to choose one out of two or more alternatives, depending on the input given. For example, suppose you design a program that determines whether the given number is even or odd. An even number is a number divisible by two without a remainder, whereas an odd number is divisible by two with a reminder. We can describe the problem with the following mathematical equation which is called expression in python.

- num % 2 = 0 ➲ num is Even, for num any integer number.
- num % 2 ≠ 0 ➲ num is Odd, for num any integer number.

A given integer number is either even or odd (i.e. two possible options), but this can be decided based on the remainder after dividing the number by two. Python provides conditional or branching statement to implement such solutions. Before starting conditional statements, it is important to learn conditional expressions in Python.

Conditional Expression

Conditional expressions are statements that use Boolean operators (such as AND, OR, NOT) or comparison operators (such as >,<. ≠). Like in mathematics, comparisons of two values in Python are described with comparison signs used in mathematics. Table 6.1 below presents comparison expressions in Python and Mathematics.

Table 6. 1 Conditional Expression

Description	Comparisons in Python	Comparisons in Mathematics
Equals	a == b	a = b
Not Equals or inequality	a != b	a ≠ b
Less than	a < b	a < b
Greater than	a > b	a > b
Less than or equal to	a <= b	a ≤ b
Greater than or equal to	a >= b	a ≥ b

The comparisons in Table 6.1 are also called conditional expressions that correspond to the Boolean values - true or false. For example, 3==4 results in false (as 3 is not equal to 4). Note that operators used in python's expressions are slightly different from operators used in Mathematics (e.g. == vs. =, != vs. ≠). Python conditionals can be used in several ways, most commonly in 'if statements' and loop statements.

Activity 6.2 Work in group

1. Write the python equivalent expression for the following mathematical equation and determine its output.
    a. 21 ≠ 21
    b. 97 ≥ 99
    c. Price = 135
2. Write a python expression to describe the following statement.
    a. Age is less than 12.
    b. Student mark is equal to 95.
    c. Weight less than or equal to 45.5.
3. What is the output of the following expressions (use python IDE to confirm your answer)?
    a. 19 != 20
    b. age = 22 age ==22
    c. 'zenash' == 'zehara'
4. What is the difference between height = 2.5 and height == 2.5 in python?

Conditional or branching statements

Conditionals are statements in a program in which the program decides at runtime whether some parts of the code should or should not be executed. From the earlier even/odd example, a given integer number can be even or odd, but this can be decided based on the remainder after dividing the number by two. Therefore, first:

Decide whether (num % 2 == 0) is or is not true.

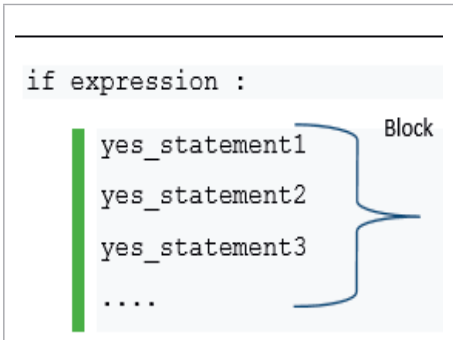
If it is true, then the number is even.

If it is not, then the number is odd.
The evaluation of the conditions (i.e., num%2==0) is a test that can be checked to see if it is true or false; as a result, executing one thing when the condition is true, or something else when the condition is false.

In Python, a conditional statement is written using the ***if*** keyword. Like many other programming languages, Python provides various versions of branching statements

that can be implemented using the ***if*** clause: simple if statement, if..else statement, and if..elif…else statement.

if statement: this is the simplest implementation of a conditional statement that decides only one part of the program to be executed if the condition is satisfied or ignored if the condition fails. The condition is an expression that is either true or false. If the condition (expression) is true, then do the indented statement(s) (See Figure 6.1). If the condition is not true, then skip the indented statements.



Evaluation of the expression is either true or false. If it is true, then the indented statement(s) is or are executed, otherwise skip the indented statements. This group of statements is called a block.

Figure 6. 1 Syntax for if…statement

Example – simple if statement

```
age = int(input("How old are you?"))
if age>0:
    print("You are", age, "years old")
    print("Thank you, nice to meet you.")
```

If your input is 1, the expression ***age>0 (i.e. 1>0)*** is true. Therefore, the two indented statements are executed, and they display their contents on the screen. However, if the input is -1, the expression age>0 (i.e. -1>0) is false. Therefore, the program ignores the indented statement.

Notes

- The expression (if age>0:) must end with colon(:). Missing colon at the end of expression is a syntax error.

Activity 6.3

- Based on the example presented under the simple if statement above:
  a. What is the output if you write age 15?
  b. What is the output if you write age 0? Discuss in pair.
  c. What is the output of if the second print statement is unindented? Discuss in pair.

Notes

- You can have any statement before or after the 'if statement' which is not considered as part of the 'if statement' and is executed as normal. However, the 'if expression' must follow at least one statement to be executed, otherwise it is a syntax error. See Figure 6.2 for syntactically correct and incorrect if statements.

Example code	Output description
`if(10 < 0):` `  print("Invalid input")`	No error, empty output
`if(10 < 11):` `  print("Invalid input")`	No error, output `Invalid input`
`if(10 < 11):` `print("Invalid input")`	Error, if clause must have at least one statement.

`if(10 < 11):`
`  print("Invalid input")`
`print("Try again")`
	`= RESTART: C:/Users`
	`Invalid input`
	`Try again`
	The last statement is not part of 'if statement', so execute as normal.

Figure 6. 2 syntax error in if statement

Activity 6.4

1. Modify age in the example above to check if the age is greater than 18, and then print 'You are Adult'
2. The following code has an error. Which part of the code does cause the error?

```
print("Hello student")
age = int(input("How old are you?"))
if(age>0):

print("You are", age, "years old")
print("Thank you, nice to meet you.")
```

3. If you type -1 and 1, what is the output of the following program? What do you noticed about the first and the last statements?

```
print("Hello student")
age = int(input("How old are you?"))
if(age < 0):
    print("Invalid input")
print("Thank you, good bay.")
```

if…else statement: The *'if...else statement'* provides two alternative statements or blocks: one following the '*if expression'* and another following the '*else* clause'. '*if…else'* allows us to specify two options: one which is executed if a condition is true (*satisfied*), and another which is executed if the condition is false (not satisfied) (See Figure 6.3 below).

if expression :
  yes_statement1
  yes_statement2
  ....
else :
  no_statement3
  no_statement4
  ....
	The interpretation of the syntax is that if the ***expression*** evaluates to ***true*** all the yes statements under **expression** are executed, and if ***expression*** evaluates to **false** all the no statements under **else** clause are executed.

Figure 6. 3: Syntax for if. else statement

Now let us see an example that makes use of the above if…else syntax. Predict what the following program prints.

As the value of age is 18, the program outputs the first print statement (See in the example below that error: reference source not found).

```
age=18
if(age > 19):
    print("You are adolescent")
else :
    print("You are children")
```

Output:

```
You are children
>>>
```

Figure 6. 4 Implementation of If Else Statement - Code (left) and Output (right)

The 'if...else statement' can make a choice between two alternative actions. Thus, the even/odd problem discussed earlier can be accomplished with the following python statement:

```
num = int(input("Input integer number"))
if  num % 2 == 0:
    print(num, " is Even number")
else:
    print (num, " is Odd number")
```

In the above program, first the expression num % 2 == 0 is evaluated (checked). If num % 2 is true (condition satisfied), then the statement following (print (num, "is Even number") is executed. Otherwise, which means the condition num % 2 == 0 has not satisfied (or false), then the statement following else (print(num, " is Odd number") is executed. Now it is time to learn additional syntax rules in python.

Syntax in a python program is a set of rules that defines how a python program is written and interpreted. For instance, the expression of an 'if statement' must be followed by a colon (:). Some programming languages require expressions to be enclosed in parenthesis, but in python it is optional. So in python, "if num % 2 == 0:" or "if (num % 2 == 0):", this complies to Python's syntax rule.

Another important syntax in python is indentation. In Figure 6.3, it is noted that the 'yes_statements' and 'no_statements' must be indented. Indentation refers to the spaces at the beginning of a code line. While in other programming languages the indentation in code is only for readability, the indentation in Python is very important.

Python uses indentation to indicate a block of code. For example, if you ignore the indentation in the example 'if statement' above, and begin the yes_statements or

the no_statements without indentation, the code generates an error. Of course, the number of spaces is up to you as a programmer, but it has to be at least one.

Notes

- Use the same number of spaces in the same block of code; otherwise, Python gives you an error (See Figure 6.5, the last if statement). These are important concepts when you write control statements.

`if (10 > 5)` `  print("Ten is greater than five", 10>5)`	It has error, missing colon(:) at the end of expression
`if (10 > 5):` `  print("Ten is greater than five", 10>5)`	It has no error, runs correctly. Enclosing expression with parenthesis is optional.
`if 10 > 5:` `  print("Ten is greater than five", 10>5)`	It has no error; it is a properly indented block.

`>>> if 10 > 5:`
`print("Ten is greater than five", 10>5)`
`SyntaxError: expected an indented block`
`>>>`
	It has an error, the block is not indented.

`>>> if 10 > 5:`
`  print("Ten is greater than five", 10>5)`
`  print("Five is less than ten!", 5<10)`
`SyntaxError: unexpected indent`
`>>>`
	It has an error because of, an inconsistent indentation in the same block.

Figure 6. 5 Illustrate Syntax rules in Python 'if statement'

Activity 6.5

1. Based on the code in Figure 6.4,
    a. What is the output if the age is 20?
    b. What if the number 0 is stored in age?
    c. When do you get the output "You are children"?
2. A group activity. Write a Python program that read two numbers from the keyboard and compute division of the first number by the second. Apply appropriate if statement to check that the second number is different from zero, otherwise print division by zero is not allowed. Do in pair

if...elif…else statement: Assume a problem that could have many options to choose from or require several conditions. For example, you want to develop a program that will print 'Excellent, Very Good, Good, Satisfactory, or Fail' based on the student mark. In such situations, Python allows you to add any number of alternatives using an ***elif*** clause after an ***if*** and before an ***else*** clause.

The **elif** is a keyword in Python to say ***"if the previous condition(s) are not true, then try this condition"***. The ***else*** keyword catches anything that is not caught by all the preceding conditions. The general syntax for 'if…elif…else' is given in

Figure 6.6 below.

```
if expression1:
    statement1
    statement2
    ...
elif expression2:
    statement3
    statement4
    ...
elif expression3:
    statement5
    statement6
    ..................
else:
    statement7
    statement8
    ...
```

Figure 6. 6 If…elif…else Syntax in Python

The interpretation of the above syntax is that first, each expression is evaluated one after the other, and if the expression is found to be ***true,*** all statements in that specific block are executed. Otherwise, if none of the expressions before the last 'else' statements are ***true***, the statements under 'else' are executed.

The program in Figure 6.7 contains a series of conditions using 'if…elif…else' to determine the status of the age (child, adolescent, adult, and senior adult). Since the age is 80, it does not match all conditions. Therefore, the statement under 'else' is

executed (See the output in Figure 6.7 under the 'Output' column).

if...elif...else based program	Output

```
age = 80
if (age < 13):
        print ("You are a Child!")
elif (age < 18):
        print ("You are Adolescent!")
elif (age < 60):
        print ("You are Adult!")
else:
        print ("You are Senior Adult!")
```

```
3.py
You are Senior Adult!
>>>
```

Figure 6. 7 conditional with if…elif…else statement

Another example in Figure 6.8 below demonstrates the application of a series of conditions using the '**if...elif…else'** statement to get the type of a variable. ***type***() which is a function that returns the type of the argument, in our case price. What is the type of price? It is a floating number. See the output in the right side of Figure 6.8 below.

if...elif...else based program	Output

```
price = 49.50
if (type(price) == int):
    print("The variable is type Integer")
elif (type(price) == float):
    print("The variable is type Float")
elif (type(price) == bool):
    print("The variable is type Bool")
elif (type(var1) == str):
    print("The variable is type String")
elif (type(var1) == list):
    print("The variable is type List")
else:
    print("The variable is unknown type")
```

```
The variable is type Float
>>>
```

Figure 6. 8 Implementation of If...elif...else Statement - Code (left) and Output (right)

Activity 6.6

1. Replace price =49.5 by the message = “Hello Student”, and the variable price in all expressions by the message in Figure 6.8 above. Then, discuss the output of the code.

2. Use appropriate conditional statements to write a Python program that solves the following problem:

➢ The program should prompt the user to enter her/his average mark in the last or the current semester and then print excellent, very good, good, satisfactory or fail based on the evaluation of the mark entered.

➢ What necessary to write the program code?

3. Write a Python program that calculates the four arithmetic operations (i.e. +, *, - and /).

➢ The program asks the user to enter one of the four arithmetic symbols. Then:

- It asks the user to enter two integer numbers.
- Based on the operator and numbers entered by the user, calculate and display the result.

Using ‘and’ and ‘or’ with if…statement: The ‘**and**’, and ‘**or**’ keywords are also used with ‘if...statement’ in Python. The ‘**and**’ and ‘**or**’ keywords are logical operators and are used to combine conditional statements.

‘and’ operator in if … statement:

```
#create two boolean variables

isBodyTempreatureHigh = True
isCoughing = True

if isBodyTempreatureHigh and isCoughing:
    print('Advice to test Covid 19')
else:
    print('You are healthy')
```

Output:

```
Advice to test Covid 19
>>>
```

Figure 6. 9 and operator in ‘if statement’ - Code (left) and Output (right)

Notes

- Any statement that is preceded by the hash symbol (#) is a comment and ignored by the Python interpreter. You will learn comments in Section 6.2.

In Figure 6.9 above, the evaluation of 'isBodyTemperatureHigh' and 'isCoughing' (i.e.True and True) is True. Therefore, the indented statement following the 'if' is executed.

Activity 6.7

1. Discuss the evaluation of the expression 'isBodyTemperatureHigh' and 'isCoughing', if the value is True and False (See Figure 6.9 above).
2. Test your response by substituting the value in the program.
3. Replace the 'and' operator by 'or' and discuss the output

Using 'or' operator with if… statement: The 'or' operator in Figure 6.10 is used to combine conditional statements. The three conditional statements compare the value of mySubject with 'Physics', 'Chemistry', and 'Geograpy'. The evaluation of the 'if' expression is True if one of these three conditions is satisfied. Otherwise, the expression evaluates to False, and in that case, it executes the statement under the else clause.

```
mySubject = 'IT'
subject = ['Physics', 'Chemistry', 'Geography', 'Economics']

#the use of or operator with if statement
if mySubject == 'Physics' or mySubject == 'Chemistry' or mySubject == 'Geograpy':
    print(mySubject + ' subject is in the list.')
else:
    print(mySubject + ' subject is not in the list.')
```

Figure 6. 10 Illustrates the use of 'on' Operator in if… Statement

Notes

- The 'subject' in Figure 6.10 is declared as array of string. An array is a variable used to store multiple values in one single variable. In the example, 'subject' stores four subject names or values.

Activity 6.8

- What is the output of the program in Figure 6.10?

6.1.2 Loops or Iteration Program Flow Controls

Brainstorming 6.3

- How do you represent 1 to 10 by repeating some pattern?

Most of real-world problems include some action that is repeated several number of times. For example, consider the program you designed in (Activity 6.6 Question 2) that determines the student's mark as "Excellent", "Very Good", etc. If you have 50 students in a class, then a more complete program would repeat this status determination 50 times (i.e. once for each student in the class). A program is often used to automate such repetitive tasks. A portion of program code that repeats a statement or group of statements in programming is called loop.

Loops are set of statements that run repeatedly for a fixed number of times, or until a condition is satisfied. Loop statements control a block of code to be executed iteratively or until a certain condition fails. Loops are a useful and frequently used feature in all modern programming languages. Python provides several language features to make iteration/looping easier. There are two types of loops that are built into Python: *for loops* and *while loops*. In the following section, implementation of 'for loop' is discussed and then followed by '***while*** loop' in python.

for loops: The *for loop* in python is used to iterate over a sequence. *For loop* in combination with the Python's range() function is used for counting in all kinds of ways (See *for loop* with range function at the end of this section). The *for loop* in python differs a bit from other like C or Pascal. In Python *for loop* is used to iterate over the items of any sequence including the python list, string, tuple, dictionary, etc. It is also used to access elements from a container (e.g. list, string, tuple) using a built-in function range(). The general syntax of *for loop* is as follows.

Syntax:

```
for variable_name in sequence :
    statement_1
    statement_2
    [....]
```

Figure 6. 11 For Loop Syntax in Python

The description of the above *for loop* syntax is:

Name	Description
variable_name	This represents a temporary variable that sets a new value for each iteration of the loop.
Sequence	Sequences are values that can be assigned to a temporary variable (i.e. variable_name). Values are provided using a list or a string or from the built-in function range().
Statement_1, statement_2 [..]	These represents a block of program statements. Python's syntax requires this to be indented.

for loop with range() function: The range() function returns a list of consecutive integers. The sequence of numbers starts from 0 by default, and counts by incrementing 1(by default), and ends at a specified number. It is widely used count controlled loops.

Notes

- In the syntax below, the range() function takes one, two, or three parameters. The last two parameters are optional.

Syntax	Example
range(x)	range(5)generate sequence 0 to 4
range(x,y)	range(2,5)generate sequence 2 to 4
range(x,y,z)	range(1,10,2)generate sequence 1,3,5,7,9

Figure 6. 12 Range Function Syntax with One, Two, and Three Parameters

range(x): generates a sequence of numbers from 0 to x, excluding x, incrementing by 1. Figure 6.13 demonstrates a simple counter program using range(5).

Python script	Output on IDLE Shell

File Edit Format Run Options
`for x in range(5):`
`  print(x)`
	`===== RESTART:`
	`0`
	`1`
	`2`
	`3`
	`4`
	`>>>`

Figure 6. 13 Implementation of for Loop with Range - Code (left) and Output (right)

Activity 6.9

1. The above program output is 0 to 4, not 5 (See Figure 6.13). Discuss what the reason is with a partner.
2. Write a program that prompts the user to enter an integer number from the keyboard and generate a sequence of numbers from 0 to user input.

   Hint: use input statement to accept user input, convert the number to a numeric type, and give as a parameter for the range function.

range(x, y): This generates a sequence of numbers from x to y excluding y, incrementing by 1. The program in Figure 6.14 uses range to generate a sequence of numbers from 5 to 8.

Python script	Output on IDLE Shell

File Edit Format Run Options
`for x in range(5, 9):`
`  print(x)`
	What is the output?

Figure 6. 14 Implementation of For Loop with Range Function

Activity 6.10

1. Write the output of the above program (in Figure 6.14) in the space provided.
2. Write a for loop that counts from 51 to 70.

range(x, y, z): This generates a sequence of numbers from x to y excluding y, incrementing by z. This is different from the above range function in that the increment is set by the z value (See Figure 6.15).

Python script

```
File  Edit  Format  Run  Options  Window
for x in range(9, 41, 4):
  print(x)
```

Output on IDLE Shell

```
===== RESTART:
9
13
17
21
25
29
33
37
>>>
```

Figure 6. 15 Implementation of For Loop with Range Function with Three Parameters - Code (left) and Output (right)

Activity 6.11

1. Write a program that generates a sequence of numbers from 10 to 100, incrementing by 10.
2. Write a for loop to generate 1, 4, 7, 10, 13, 16, 19, and 22.

for loop in iterable object: Now let us see an example of a *for loop* in an iterable object. Unlike the earlier example, the loop iterates while something is true. This type of loop is called a condition controlled loop.

Python script

```
File  Edit  Format  Run  Options  Window  Help
#print each item in the list regionalState
regionalState = ["Afar", "Amhara", "Benishangul-Gumuz",
                 "Gambela", "Harari","Oromia", "Somali",
                 "Tigray", "SNNPR", "Sidama"]
for region in regionalState:
        print(region)
```

Output on IDLE Shell

```
===== RESTART: C:\
Afar
Amhara
Benishangul-Gumuz
Gambela
Harari
Oromia
Somali
Tigray
SNNPR
Sidama
>>>
```

Figure 6. 16 Implementation of For Loop - Code (left) and Output (right)

The *for loop* program in Figure 6.16 is condition controlled. The 'regionalState' is a sequence that contains a list of regional states. When the *for loop* is executed, the first item (i.e. Afar) is assigned to the 'region'. The 'region' is a temporary variable that actually represents an element in the list. After this, the print statement executes, and the process continues until we reach the end of the list (i.e. Sidama), or while there is an element in the list. See the output in Figure 6.16 above on the right hand side.

Activity 6.12

Write a for loop statement that displays the following list of fruits.

- fruits = ["Mango", "Orange", "Banana", "Pineapple", "Papaya"];

Syntax: for variable_name in string

for loop can iterate through string. The string is an iterable object in python because it contains a sequence of characters. Thus, applying ***for*** *loop* in a string allows us to access the content character by character.

Python script

```
File  Edit  Format  Run  Options  Window  Help

for letter in "Renaissance Dam":
  print(letter)
```

Output on IDLE Shell

```
===== RESTART:
R
e
n
a
i
s
s
a
n
c
e

D
a
m
>>>
```

Figure 6. 17 for Loop in a String

In the above program, in each iteration, one character is accessed from the string value and stored in the letter variable, and printed on the screen. The iteration continues until the last character (i.e. m) is accessed and printed on the screen.

Activity 6.13

- Write for loop to get your name from the keyboard and print character by character.

Hint: use input statement to read your name from the keyboard.

break Statement with for: The term break is a keyword in python. With the break statement, you can stop the loop before it has looped through all the items:

Python script	Output on IDLE Shell

```
File  Edit  Format  Run  Options  Window  Help
programming_language =["Java", "Python", "C++",
                       "C Sharp", "Java Script"]
for language in programming_language:
  print(language)
  if language == "Python":
    break
```

```
===== RESTART:
Java
Python
>>>
```

Figure 6. 18 for Loop with break Statement – Code (left) and Output (right)

In the above program, the loop exits when the value of 'language' is 'Python'.

Activity 6.14

1. Modify the program in Figure 6.17 (for loop in string) above to stop the loop if the value of letter is white-space.

   Hint: use 'if statement' to check if letter == ' '

2. What is the output of Figure 6.18 if you replace the 'if statement' by if language== "HTML"?

continue keyword with for loop: The term continue is a keyword in python. With the **continue** statement, you can stop the current iteration of the loop, and it continues with the next.

Python script	Output on IDLE Shell

```
File  Edit  Format  Run  Options  Window  Help
programming_language =["Java", "Python", "C++",
                       "C Sharp", "Java Script"]
for language in programming_language:
  if language == "C++":
    continue
  print(language)
```

```
===== RESTART:
Java
Python
C Sharp
Java Script
>>>
```

Figure 6.19 for loop with continue statement - Code (left) and Output (right)

Activity 6.15

- Modify the above statement to escape “Java” and “C++” using continue keyword.

Hint: use the operator to combine the condition if language ==”Java” or language==”C++”.

while Statement: The *while statement* in Python supports repeated execution of a statement or block of statements that is controlled by a conditional expression. The general syntax for the *while statement* is:

Syntax:

while expression: Statement_1 Statement_2 […]	Note that in python expression must end by colon(:). Statements under the **while** must be indented.

Figure 6. 20 while loop syntax

The *while loop* runs as long as the condition (expression) evaluates to True and executes the program block (statement_1, statement_2 …). The expression is checked every time at the beginning of the loop and the first time when the expression evaluates to False, the loop stops without executing any remaining statement(s).

Python script	Output on IDLE Shell

File Edit Format Run
`count = 1`
`while count < 5:`
`  print(count)`
`  count += 1`
	`===== RESTART:`
	`1`
	`2`
	`3`
	`4`
	`>>>`

Figure 6. 21 Implementation of While Loop - Code (left) and Output (right)

The *while loop* above prints the value of count as long as the count is less than 5. The last statement (count += 1) is important to increment the value of count at every iteration and eventually terminate the loop.

Notes

- Don't forget to increment count(i.e. count +=1 in Figure 6.21); otherwise, the loop continues forever.

Activity 6.16

- Rewrite the problem described in Activity 6.11 using a 'while' statement.

Hint: Set the count to 10, and increments count by 10.

break and continue keywords with while loop: With the **break** statement, you can stop the loop even if the **while** condition is true. It causes the loop to quit even before reaching the last iteration. The loop in the program below terminates when the value of count is 5 (in the 5th iteration).

Python Script	Output on IDLE Shell

File Edit Format Run
`count=1`
`while count< 10:`
`  print(count)`
`  if count == 5:`
`  break`
`  count += 1`
	`===== RESTART:`
	`1`
	`2`
	`3`
	`4`
	`5`
	`>>>`

Figure 6. 22 Implementation of While Loop with break Statement - Code (left) and Output (right)

As you have learned in **for loop** above, the **continue** statement, causes the current iteration to stop, and continues with the next.

Python script	Output on IDLE Shell

File Edit Format Run Options Window Help

```
# read input from the user
escape = int(input("Enter iteration to escape"))
count = 1
while count < 10:
  count += 1
  if count == escape:
    continue
  print(count)
```

```
===== RESTART: C:\Users\Dag
Enter iteration to escape7
2
3
4
5
6
8
9
10
>>>
```

Figure 6. 23 Implementation of While Loop with continue Statement - Code (left) and Output (right)

The above program is supposed to iterate 9 times but prints 8 values by escaping print when the value of count is equal to escape.

Activity 6.17

- If the user input is 8, what is the output of the above (in Figure 6.23) program?

6.2. Comments in Python

Brainstorming 6.4

- Have you ever noticed a comment in a program so far? What is its purpose?

Comments are descriptive texts that exist in program source code and are ignored by a compiler and interpreter. Comments are not executable statements or part of the program. Using comments, a program can make code more readable for other developers as it provides some information or explanation about what each part of a program is doing. Depending on the purpose of your program, comments can serve as notes to yourself or reminders.

In python, comments are denoted by the hash symbol (#). Anything after the # symbol is ignored by the interpreter (See the above example program code with the # symbol). Comments can be given in a single line or may take multiple lines. In any case, all comment lines should start with the special character (#). In general, you can use comments to describe your program code, to make the code more readable, or to prevent the execution of some parts of the code while testing the code. Examples of each are given below.

Creating comment: In Python, comments start with a hash symbol (#), and the Python interpreter ignores them.

```
File  Edit  Format  Run  Options  Window  Help
#This is my first comment
print("Hello, Ethiopians!!!")

pi = 3.14  # this is inline comment initialize pi
```

Figure 6. 24 Illustration of Comments in Python Script

In the above example, the two comments (i.e. ones that start with #) are ignored while executing the program.

Commenting part of the code: Comments can also be used to comment out parts of the code that you do not want to execute while compiling or running a program. The program in the following example has two fragments of code. The second fragment of code (next to line pi = 3.14) is commented out to prevent that part of the code from execution. IDLE provides features to comment out (uncomment) parts of code in your source program.

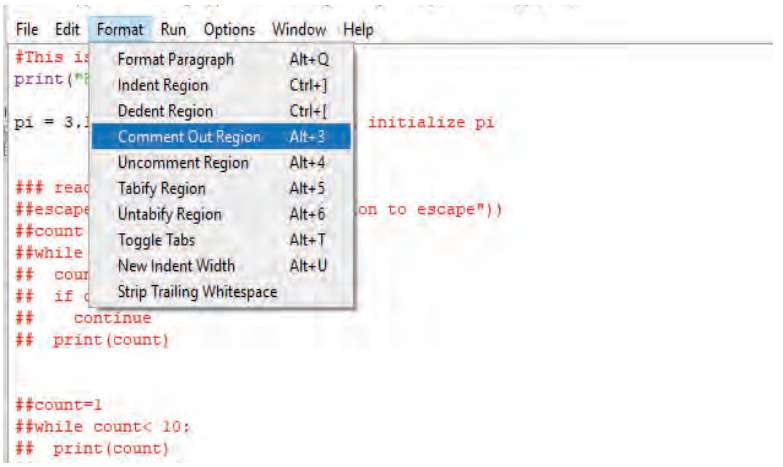
```
File  Edit  Format  Run  Options  Window  Help
#This is my first comment
print("Hello, Ethiopians!!!")

pi = 3.14  # this is inline comment initialize pi

### read input from the user
##escape = int(input("Enter iteration to escape"))
##count = 1
##while count < 10:
##  count += 1
##  if count == escape:
##    continue
##  print(count)
```

Figure 6. 25 Illustration of Commenting out Part of Code

Commenting in IDLE: In IDLE, commenting features are found under the 'Format' menu. Select part of the code you want to comment out, then click on the 'Format' menu, and from the list, select 'Comment Out Region', or you can use the short cut key 'Alt+3'. In order to uncomment, select the part of code you want to uncomment. Then click on 'Format' menu, and from the list, select 'Uncomment Region', or you can use short cut key 'Alt+4'.



To comment out

Select part of code

Click on Format menu

From the list, select

Comment Out region or use Alt+3

Figure 6. 26 Illustrate Commenting Tool in IDLE

```
File  Edit  Format  Run  Options  Window  Help
### read input from the user
##escape = int(input("Enter iteration to escape"))
##count = 1
##while count < 10:
##   count += 1
##   if count == escape:
##      continue
##   print(count)


count=1
while count< 10:
  print(count)
  if count == 5:
    break
  count += 1
```

Figure 6. 27 Illustrates the First Part of the Code Commented out

Activity 6.18

1. Comment on the second part of the program in the above figure and run the program. Then, write the output.
2. Uncomment the first part of the program in the above figure and run the program. Then write the output.
3. Add comment that describes the second part of the program.

When you experience errors after writing new lines of code, then you might comment on a few of them to see if you can troubleshoot the precise issue.

6.3. Python Interpreter

Brainstorming 6.5

- Describe the python program writing process.

In this section, program writing and running are discussed in more detail.

Python is an interpreted programming language, not a compiled one although compilation is a step in Python interpreter processes. Compilation is simply a translation that generate bytecode that you will more in the next paragraph. The interpreter is the program that is responsible for executing each line of statements sequentially and runs the Python code or script which refers to a simple program stored in a plain text file. You can think of the interpreter as a layer of a program that works between your program and your computer hardware to get your code running. Python has a wide range of interpreters including IDLE, CPython, Jython, PyPy, and IronPython and online Python editors. Your Python source code is always run by one of these interpreters depending on the Python installation you use. Therefore, first, make sure that you have correctly installed one of these interpreters on your computer.

Links

Read your Grade 11 IT Textbook for more on Python IDE

Steps in the Python Interpreter Process

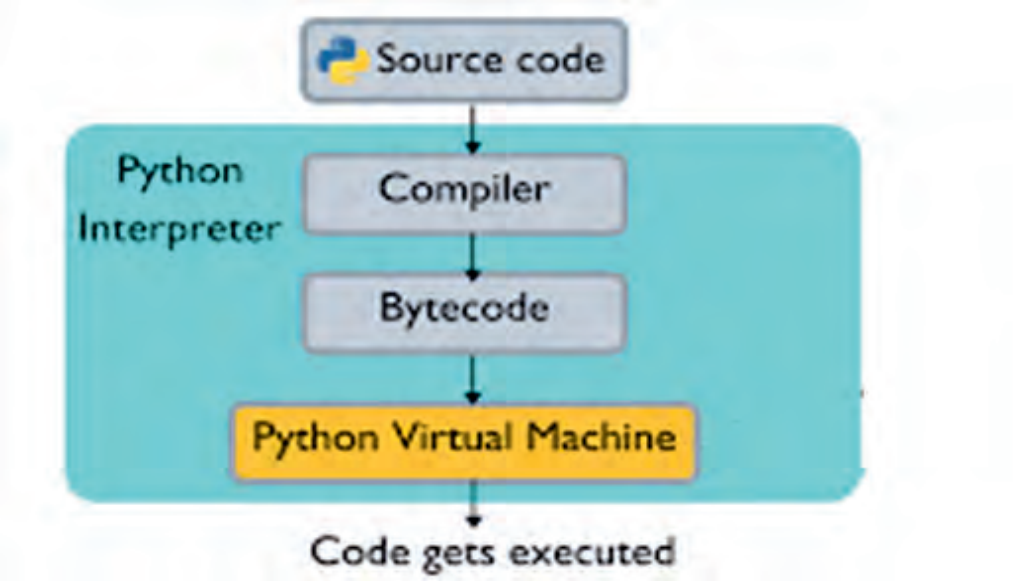


Figure 6. 28 Python Interpreter Source Code Compilation Process

Given that you write and save your Python code with the extension **.py**, first your source code (.py file) compiles into a format called **bytecode**. Compilation is a simple translation process that generates a bytecode. In Python a compiled code is usually stored in a file with the extension of **.pyc,** which can update when necessary. Then, the **bytecode** (.pyc file) is loaded into the Python runtime and interpreted by a **Python Virtual Machine (PVM)**. The PVM is the Python runtime powerhouse. The PVM is a piece of program that reads each instruction in the **bytecode** one by one and executes whatever operation is indicated. **Bytecode** interpretation is automatic, and the PVM is just part of the Python system that you have installed on your machine. The PVM is the component that truly runs your scripts. Running your program is just the last step of what is called the Python interpreter process.

Activity 6.19

- Describe the steps involved in the python interpreter process.

6.4. Testing and Debugging Programs

Brainstorming 6.6

- Why do we need to test a program?

Once you write your program source code, you save the file and run the code to test whether the desired or expected output is generated. The program sometimes fails to run correctly because of a bug. A **bug** is an unexpected problem in your program. Bugs can appear in many forms, and some are more difficult to fix than others. Some bugs are tricky enough that you will not be able to catch them by just reading through your program. Luckily, Python IDLE provides some basic tools that help you debug your programs with ease. Debugging means, having complete control over the program execution.

Getting Started Interpreter DEBUG Mode

If you want to run your code with the built-in debugger in Python IDLE, you need to turn to debug mode feature. To do so, select *Debug → Debugger* from the Python IDLE menu bar. In the interpreter, you should see [DEBUG ON] appear just before the prompt (>>>), which means the interpreter is ready and waiting (See Figure 6.29 below).

When you run your Python file, the debugger window appears (See Figure 6.29 below):

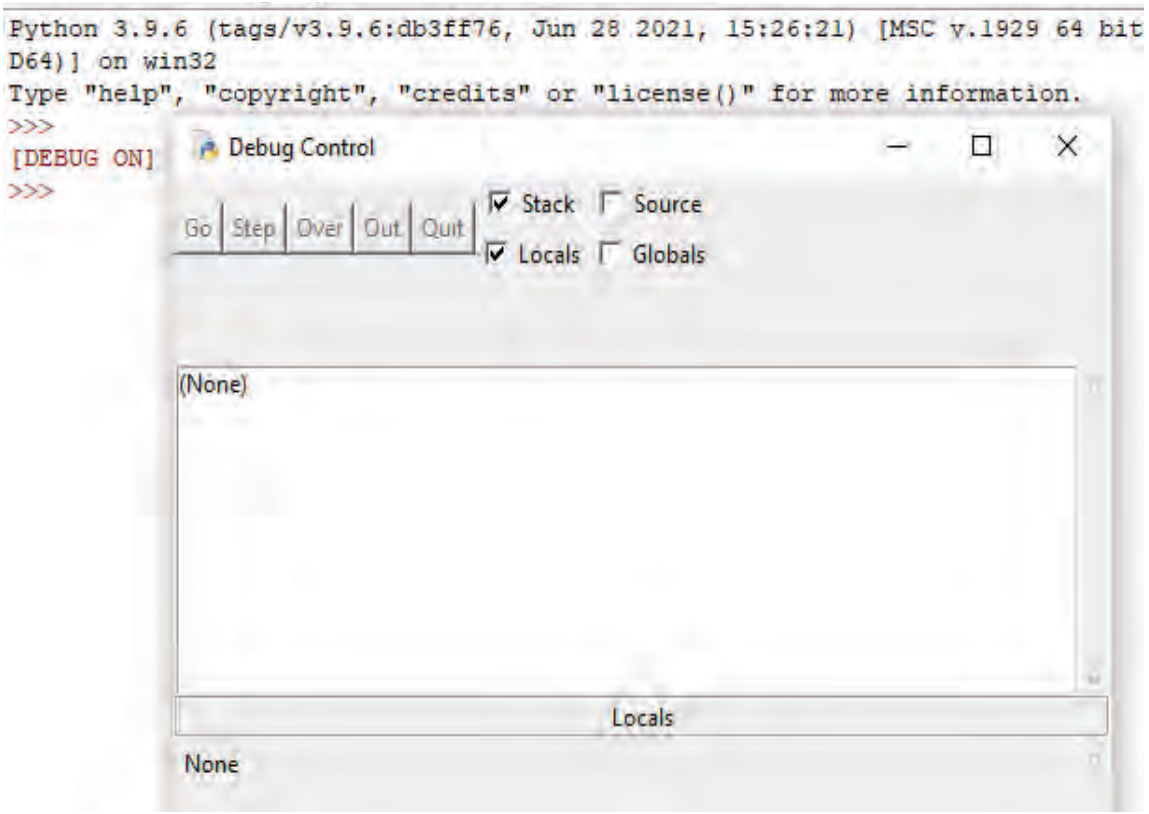


Figure 6. 29 Debug Control Window When Debug Mode On

In this window, you can inspect the values of your variables as your code executes. This gives insight into how your data is being manipulated as your code runs.

Click the following buttons to move through your code:

Go: Press **Go** to advance execution as normal until a breakpoint is encountered (or input is requested).

Step: Press **Step** to see all the internal commands that python uses to execute the current line and go to the next one.

Over: Press the **Over** option to see line-by-line execution of the program.

For our purposes, we can use 'Go' and 'Step' to trace and fix the bug in the following program. Before running the program, set a breakpoint on a statement that causes an error in your program.

A **breakpoint** is a line of code that you have selected as a place where the interpreter should pause while running your code. To set a breakpoint, right-click on the line of code that you wish

```
File  Edit  Format  Run  Options  Window  Help

num1 = input("Enter the first number to add: ")
num2 = input("Enter the second number to add: ")
print("Product ", num1 * num2)
```

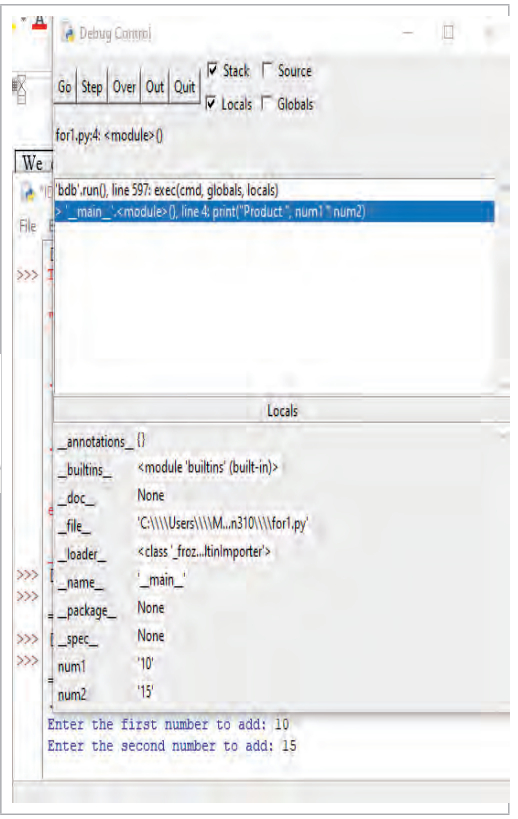
Figure 6. 30 a breakpoint set to code in the program.

to pause(See Figure 6.30 that Error: Reference source not found). This highlights the line of code in yellow as a visual indication that a breakpoint is set. You can set as many breakpoints in your program code as you like. To undo a breakpoint, right-click the same line again and select *Clear Breakpoint.*

Now run the program. Then from the DEBUG control (shown on the left) press 'Go'. This prompts you to type values for num1 and num2. Enter the numbers from the keyboard and press enter.

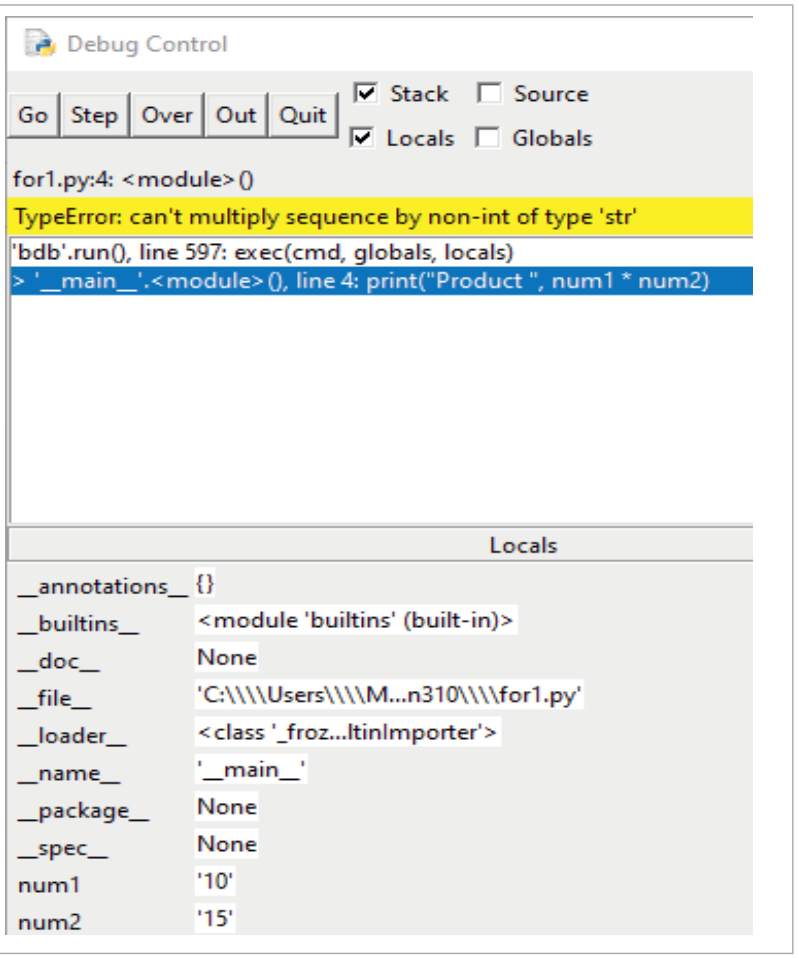
As shown below.

```
Enter the first number to add: 10
Enter the second number to add: 15
```



Now press 'Step' from Debug Control to execute line by line. This displays TypeError (highlighted in yellow). Go to the source code to fix the problem by casting the input to integer.

Then repeat the above steps to check the bug is fixed



Activity 6.20

- The following code could generate error if the value for num2 is zero. Trace and debug the error using python DEBUG control.

File Edit Format Run Options Window Help

```
num1 = int(input("Enter the first number to add: "))
num2 = int(input("Enter the second number to add: "))
print("Division: ", num1/num2)
```

Unit Summary

In this unit, you have learnt about:

- program flow control.
- the special purpose of indentation in python.
- types of flow controls in python.
- conditional or branching statements in python – if, if…else, if…elif…else, if with in/and/or.
- looping or iteration statements in python – for…loop, while…loop, for loop with range(), break, and continue.
- the purpose of comment in program source code.
- using IDLE commenting features
- steps in the python interpreter process
- testing and debugging program code – Debug Control and breakpoint

Key Terms

Program flow controls - A program flow control refers to the order of execution of the program's code. Python implement the flow control of the program through the sequential statements, conditional statements and loop statements.

Syntax in a python program - is a set of rules that define how a python program is written and interpreted.

Indentation - Indentation refers to the spaces at the beginning of a code line. Unlike other programming languages, python uses indentation to indicate a block of code.

The number of spaces is up to you as a programmer, but it has to be at least one.

Conditional statements -are statements in a program where there are points at which the program decides at runtime whether some parts of the code should or should not be executed.

Conditional statements -can be implemented by 'if statement'. Such as simple if, if…else, if…elif…else, nested…if

Iteration - This is a programming logic to automate repetitive tasks. Repeated execution of a set of statements is called iteration. Python provides various versions of for statements and while statements to implement iteration.

Continue -is a keyword which is used with for loop. With the continue statement, you can stop the current iteration of the loop, and continue with the next.

Break - is a keyword which is used with for loop. With the break statement, you can stop the loop before it has looped through all theitems.

Comments - are descriptive texts in program source code that are ignored by compilers and interpreters. Using comments, programs can make code more readable for humans as it provides some information or explanation about what each part of a program is doing or about.

Interpreter is a special program that executes instructions written in a programming language. It can either execute the source code directly or translate the source code in a first step into a more efficient representation, and execute this code.

A **bug** is an unexpected problem in your program. Bugs can appear in many forms, and some are more difficult to fix than others.

A **breakpoint** is a line of code that you have identified as a place where the interpreter should pause while running your code.

Review Questions

Part I: Write True if the statement is correct and False if it is incorrect.

1. There are three program logics: sequential, conditional (branching), and iterative (looping).
2. Indentation in python is used for code readability only.
3. In python, looping can be implemented using for and while statement.
4. The order in which the program code execution is determined is referred as flow control.
5. In python, we can apply different indentations in a program.
6. Condition (expression) in the conditional statement must evaluate to either true or false.
7. In while statement, the loop continues as long as the condition is satisfied.
8. A comment is an executable statement.
9. Interpreter can either execute the source code directly or translate the source code in a first step into a more efficient representation and execute this code.
10. A bug is a syntax error in your program.

Part II: Match the items given under column B with associated items in column A

A	B
1. Program flow control	A. if, if…else, if…elif..else
2. Bytecode(.pyc file)	B. interactive interpreter
3. Breakpoint	C. a>b, a==b
4. Indentation	D. writes, compiles, tests and runs a program
5. Branching statement	E. line interpreter pauses while running
6. Nested loop	F. Loop inside another loop
7. Condition(expression)	G. Order of code execution
8. IDE	H. Indicates block of code
9. Read-Eval-Print Loop (REPL)	I. Any line preceded by the hash(#)
10. Comment	J. Python Virtual Machine(PVM)

Part III: Choose the correct answer from the given alternatives.

1. All are a program flow controls except____.

   A. sequential statement B. branching statement
   C. looping D. comment

2. The statement range(9) generates a sequence of numbers from_____.

   A. 0 to 9 B. 0 to 8
   C. 1 to 9 D. 1 to 8

3. One of the following cannot be the condition(expression) of 'if statement':

   A. a = b B. a !=b
   C. a >=b D. a == b

4. Which of the following indentation is not valid in python?

   A. if 5:
      print('five')
   B. if a>b:
      print('greater than')
   C. if 4 > 5: print('false')
   D. if a:
      print('yes')

5. Compiled source code is stored in_____.

   A. .pyc file
   B. .py file
   C. Pvm file
   D. Compiler

Part IV: Code Writing

1. Use branching statement to write a program that reads student mark, and determine pass if the mark is greater or equal to 50; otherwise, print fail.
2. Given letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ":
   a. Write a for loop statement to print a character per line in each iteration.
   b. Repeat the for loop but the character should print in a single line.
3. Write a program based on the following requirements.
   a. Prompt the user to input an integer number from the keyboard.
   b. Then print all sequences of numbers from 1 to the input number.
   c. Display the data type of the variable before and after the reassignment.
4. Write a program that performs the arithmetic operation (+, -, *,/).

   Hint: Ask the user to enter the operation to perform(1 for addition, 2 for subtraction, 3 for multiplication, and 4 for division;Otherwise, print invalid input.