جامعة بوليتكنك فلسطين
Palestine Polytechnic University

2024-2025 | 5055

الحاسوب وأساسيات البرمجة

Computer and Programming Principles

Prepared By
Eng. Yousef Adnan Salah

تأليف وإعداد
المهندس يوسف عدنان صلاح

This material developed under the objectives of FESTEM project funded by EU.
www.festem.ps

بسم الله الرحمن الرحيم

الحمد لله الذي بنعمته تتم الصالحات، والصلاة والسلام على خير خلق الله محمد وعلى آله وصحبه أجمعين.

تم إعداد هذه النسخة من المادة التعليمية لمساق "الحاسوب وأساسيات البرمجة" في جامعة بوليتكنك فلسطين، بهدف تلبية متطلبات مخرجات هذا المساق الأساسي في الجامعة، وتزويد الطلبة بالمفاهيم والمهارات التقنية التي تساهم في بناء شخصيتهم الرقمية، وتجعل منهم أكثر وعياً واستعداداً للانخراط في عالم تكنولوجيا المعلومات وبرمجة الحاسوب.

وبالرغم من الجهود المستمرة في تطوير المساق سنوياً لمواكبة أحدث المستجدات الرقمية، لكنني أؤمن أن الكمال لله وحده. لذلك فإنني أرحب دائماً بتلقي ملاحظاتكم وتوجيهاتكم لتحسين هذا المحتوى وجعله أكثر تكاملاً.

لا تترددوا في إرسال آرائكم عبر البريد الإلكتروني: ysalah@ppu.edu

مع أطيب التمنيات بالتوفيق والنجاح،

م. يوسف صلاح

2025

جامعة بوليتكنك فلسطين

INTRODUCTION TO COMPUTERS

Prepared by:
Eng. Yousef Salah

This material developed under the objectives of FESTEM project funded by the EU.
https://festem.ps/

Co-funded by the Erasmus+ Programme of the European Union

CHAPTER ONE
INTRODUCTION TO COMPUTERS

PREFACE:

This chapter highlights key concepts and terminologies related to computing and information technology. It explores the fundamental functions of computer systems and their significant impact on our daily lives. Additionally, we will examine the various types of computing machines used worldwide, distinguishing between them based on different criteria.

INTENDED LEARNING OUTCOMES:

1) Teach the fundamentals of computers and computer terminologies, particularly with respect to personal computer hardware and software.
2) Give students an in-depth understanding of how computers are essential in our daily practices.
3) Recognize the main functions of computing, and gain the awareness of trendy computing sciences.
4) Present the foremost types of computer machines, and give students enough knowledge about the traits of each one.

FURTHER READING:

1) Discovering Computers ©2018: Digital Technology, Data, and Devices.
2) Computing Essentials 2017-McGraw-Hill (2017) Daniel O'Leary, Linda I. O'Leary, Timothy J O'Leary .

WHAT IS A COMPUTER?

- A Computer is a digital machine that operates under a set of instructions and rules, so that it will be capable of inputting data, performing processing, producing results, and probably storing results for future use.

- The computer can be viewed from two perspectives:

  - Hardware: the physical parts of the computer, which consist of electrical, electronic and mechanical parts that compose the computer machine.

  - Software: data and instructions at which when executed will operate and manage the computer hardware.

ROLE OF COMPUTERS IN OUR LIFE:

Computers are nowadays playing a vital role in every activity in our life. For instance, you may use computers and their applications in:

1) Education and Scientific Research
2) Business, Shopping and Marketing Fields
3) Healthcare sector
4) Banking and Financial contracts
5) Industries
6) Government offices
7) Entertainment
8) Home and Building Management Systems
9) Weather Forecasting
10) Social Communication

Activity:

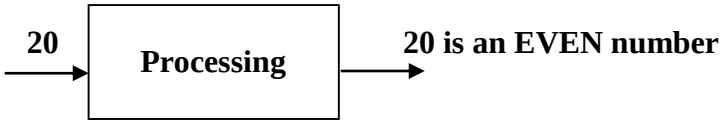
Have you ever created a professional profile on LinkedIn Network!
It is the time now to do so and to expose yourself to the world of professional networking on the Internet. Visit: *https://www.linkedin.com* and start creating your existence there.

DATA, INFORMATION, AND KNOWLEDGE:

- **A computer processes data to produce information:**

  - Data are raw facts and items that don't convey any meaning.
    - Data may be: numbers, characters, symbols, texts, sounds, images,….
    - Examples: 20, "Ahmad", 'A', 986.35 ….

  - Information: processed data that arranged in a way that looks useful and conveys meaning in an understandable way.



- A level that comes after information called "Knowledge". Knowledge is the understanding of information about a subject that you get by experience, study, perceiving, discovering, or learning.

- Nowadays, computers make many decisions without human intervention. That is, they receive data as input and they process it in a manner that produces decisions and actions as output. This leads to the foremost field in computing which is called "Artificial Intelligence".

- Artificial Intelligence (AI) is to have intelligent machines that mimic human's minds, based on the perception of its environment, and based on a set of rules that enable the machine to take decision properly, while it learns by experience.

BASIC COMPUTING TERMINOLOGIES

- Computer Science (CS):
  CS involves the study of computers and how they work, including how to create software, solve problems using algorithms, and manage data. It involves programming, understanding how hardware and software interact, and learning to think logically to develop efficient solutions for real-world tasks.

- Information Technology (IT):
  IT is the use of computers and software to manage, store, and share information. It focuses on making sure technology systems work well, including hardware, networks, and data security, to help businesses and organizations run efficiently.

- ICT is an acronym for "Information and Communication Technology" which revolves *around the use of technology to communicate and share information. It includes tools like computers, the internet, mobile phones, and other devices that help people connect, exchange messages, and access information.*

Activity:

Visit: *https://www.wikipedia.org* and start searching for more details of the aforementioned definitions.

COMPUTER TYPES:

Computers may be classified into:

- Personal Computers (PC)
- Smart Devices (Mobile Devices)
- Servers
- Mainframes
- Supercomputers
- Embedded Computers

- Before delving into the details of each type, we ought to ensure that any system capable of inputting data, processing or manipulating it, generates results, and storing these results is a *Computing Machine*.

- We'll classify computers based on different factors like:
  - Uses of the computer
  - Size
  - Number of users at the same time
  - Operating System (OS): The software responsible for the operation and management of the computer.
  - Price
  - Performance

1 Personal Computers (PC)

- A Personal Computer is a small computer designed to be used by one person at a time. A PC is available in variety of shapes and sizes like:
  - Desktop Computers: designed to fit on your desk.
  - Portable Computers: comes with built-in screen, keyboard, and battery. They are movable computers from place to place. Laptops are considered personal computers.
- Suitable for every day computing, browsing the Internet, creating documents, playing games, sending emails and the like.
- Cheap price and suitable performance.

2 Smart Devices, Mobiles and Tablets

- These devices are a major part of everyday life, combining computing power with portability and connectivity.
- Smart devices are gadgets that connect to the internet, interact with other devices, and often operate autonomously to perform tasks.
- Examples of Smart Devices:
  - **Smartphones**: Mobile phones with advanced capabilities like internet access, apps, and sensors.

- **Smartwatches**: Wearable devices like Apple Watch or Fitbit that track health, show notifications, and more.
- **Smart Home Devices**: Devices like **smart speakers** (Amazon Echo), **thermostats** (Nest), or **lights** that can be controlled via voice commands or apps.
- **Smart TVs**: Televisions that can stream content, run apps, and connect to other devices without additional hardware.

– Prices range from very cheap to expensive, depending on model, with special Operating Systems.

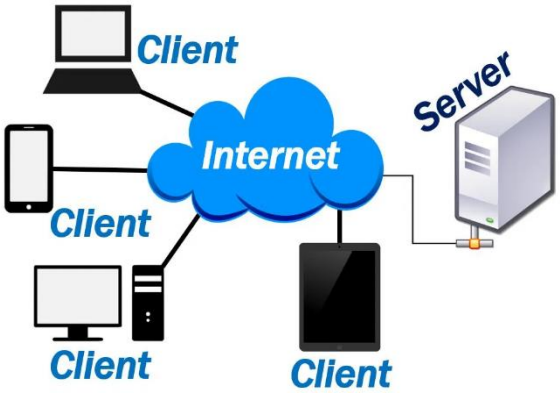
– Mobile devices include mobile phones, smartphones (mobile phones that access the Internet and run apps), and small tablet devices (such as iPads and Android tablets).

3 Servers

– A ***Server*** is a powerful computer used to host programs and data for a network.

– It provides services for other computers called *clients*.

– Usually have more processing power than Personal Computers.



– These computers run 24/7.

– Examples: Email Servers, Web Servers, Game Servers, File Servers, and application servers.

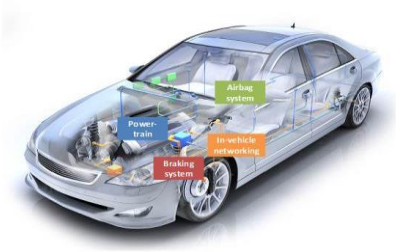
4 Mainframes

- Mainframe computers are very powerful and large computers used mainly by big organizations. They handle huge amounts of data and perform many tasks at once, such as running complex applications and managing large databases.
- Mainframe computers are capable of great processing speeds and data storage.
- Large size, like the size of a closet.
- More expensive than servers.
- Used by Banks, Hospitals, Universities, and research centers that need to manage large amounts of centralized data.
- Can handle hundreds to thousands of users at the same time.

5 Supercomputers

- A Supercomputer is the fastest, most powerful computer, and the most expensive.
- Very large size (size of a room).
- Supercomputers are typically used to process massive amounts of data.
- For example, large-scale simulations and applications in medicine, aerospace, automotive design, online banking, weather forecasting, nuclear energy research, and petroleum exploration.

6 Embedded Computers



- An *embedded computer* is a special-purpose computer that functions as a component in a larger machine or device.
- Found today in most electronic devices like Cars, air planes, home appliances (dishwashers, microwave ovens and refrigerators), Game Consoles (Wii, PlayStation), Printers, Digital Cameras, televisions, and more.
- Designed and programmed to operate and perform specific tasks in real time.

Activity:

What is IoT? Is there any relevance between IoT and embedded computer?

Activity:

In one sheet, and based on what discussed in the class, differentiate between the 6 general types of computers.

CHAPTER QUESTIONS:

Q1) State whether each of the following statements is **TRUE** or **FALSE**.

	Statement	TRUE / FALSE
[1]	Computers process information to create data.	
[2]	Hardware consists of a series of instructions that tells the computer what actions to perform and how to perform them.	
[3]	Because embedded computers are components in larger products, they usually are small-sized devices and have limited hardware.	
[4]	Data conveys meaning to users, and information is a collection of unprocessed items, which can include text, numbers, images, audio, and video.	
[5]	Embedded computers generally are special purpose computers.	

Q2) Choose the best answer:

1	LinkedIn is a:	**A.** Type of Computer **B.** A Software **C.** A Server **D.** Professional Network
2	_____ is/are the steps that tell the computer how to perform a particular task.	**A.** Data **B.** Information **C.** Instructions **D.** Documentation
3	Although not as powerful as a supercomputer, this type of computer is capable of great processing speeds and data storage.	**A.** Laptop **B.** Mainframe **C.** Supercomputer **D.** Desktop Computer
4	The circuitry of the computer is part of:	**A.** Software **B.** Hardware **C.** Data **D.** Firmware
5	A game console is a type of:	**A.** Embedded Computer **B.** Server **C.** Mobile Device **D.** Mainframe

جامعة بوليتكنك فلسطين

COMPUTER HARDWARE

Prepared by:
Eng. Yousef Salah
Eng. Mohammad Jabari

This material developed under the objectives of FESTEM project funded by the EU.
https://festem.ps/

Co-funded by the Erasmus+ Programme of the European Union

CHAPTER TWO
COMPUTER HARDWARE

PREFACE:

This chapter provides a broad overview of the primary physical components of a computer system. It also introduces students to different types and characteristics of computer processors, memory, storage devices, and input/output devices.

INTENDED LEARNING OUTCOMES:

After completing this chapter students will be able to:

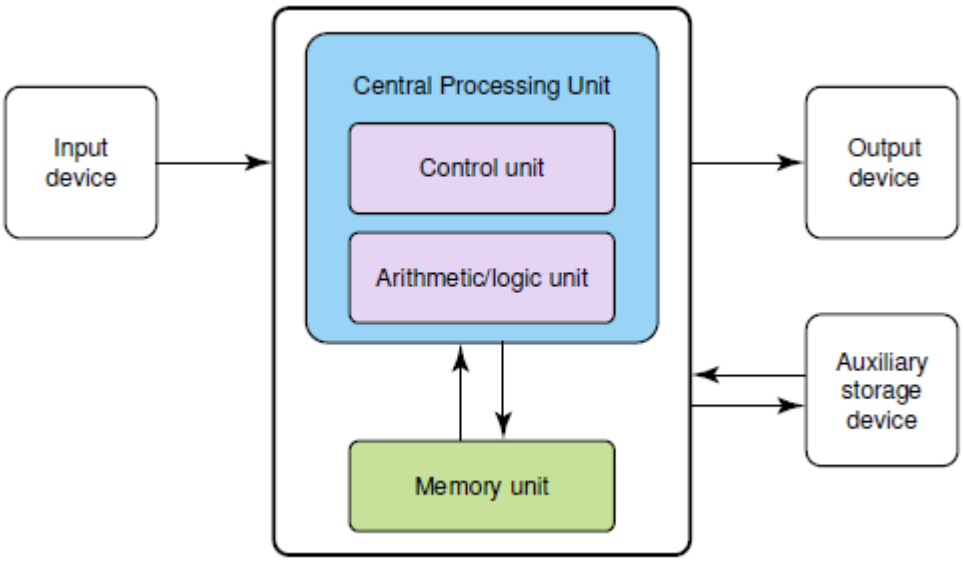
1) Introduce students to the basic hardware parts of the computing machine.
2) Illustrate the main components the computer processor made of and their functionalities.
3) Differentiate between main types of computer memory and storage media.
4) Study the metrics used in identifying memory capacity.
5) Present miscellaneous input and output devices.

FURTHER READING:

3) Discovering Computers ©2018: Digital Technology, Data, and Devices.
4) Computing Essentials 2017-McGraw-Hill (2017) Daniel O'Leary, Linda I. O'Leary, Timothy J O'Leary .

INTRODUCTION TO COMPUTER HARDWARE

- As discussed in Chapter One, computer hardware is the physical and tangible parts of the computer, which include all electronic, electrical and mechanical items inside the computer machine.
- The ***Central Processing Unit*** (CPU) is the "brain" of a computer system, containing digital logic circuitry able to interpret and execute instructions.
- ***Main Memory*** is where currently executing programs reside, which the CPU can directly and very quickly access. Main memory is volatile; that is, the contents are lost when the power is turned off.
- ***Secondary Memory*** is nonvolatile, and therefore provides long-term storage of programs and data. This kind of storage, for example, can be magnetic (hard drive), optical (CD or DVD), or nonvolatile flash memory (such as in a USB drive).
- ***Input/Output Devices*** (or **Peripherals**) include anything that allows for input (such as the mouse and keyboard) or output (such as a monitor or printer).
- ***Buses*** transfer data between components within a computer system, such as between the CPU and main memory.
- The figure below shows the main hardware components of a computer system:



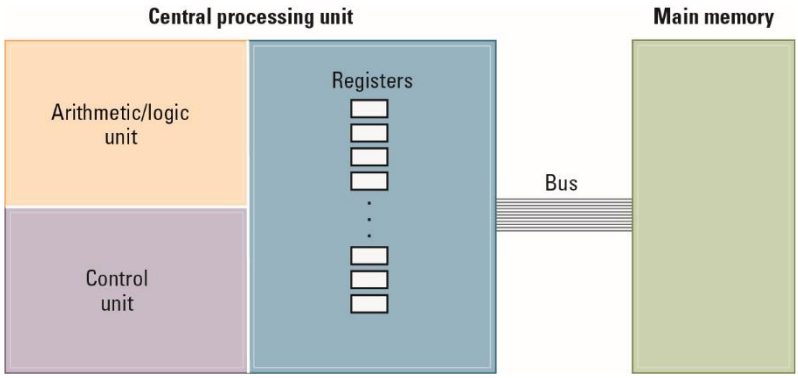
- Every computing machine mainly the following basic hardware components:
  - Processing Components
  - Memory Units
  - Storage Components
  - Input/Output Devices

Processing Components:

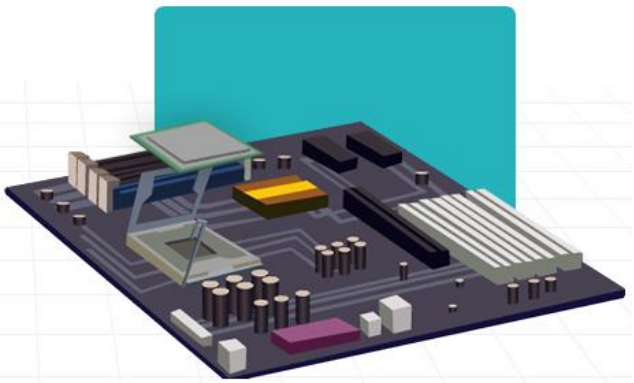
Within the processing components, data is processed and information is generated that will be displayed on the output components.

➔Processing Units

- The main component is the CPU (Central Processing Unit) or Processor.
- The processor contains the following main subparts:



- **Arithmetic and Logic Unit (ALU)**
  Performs Arithmetic and logical operations, like Addition, Subtraction, Comparing, And, OR, etc …
- **Control Unit (CU)**
  Controls all operations on CPU, and communicates with both the ALU and memory.
- **Registers**
  Registers are used by the CPU to store data and intermediary results temporarily during processing.



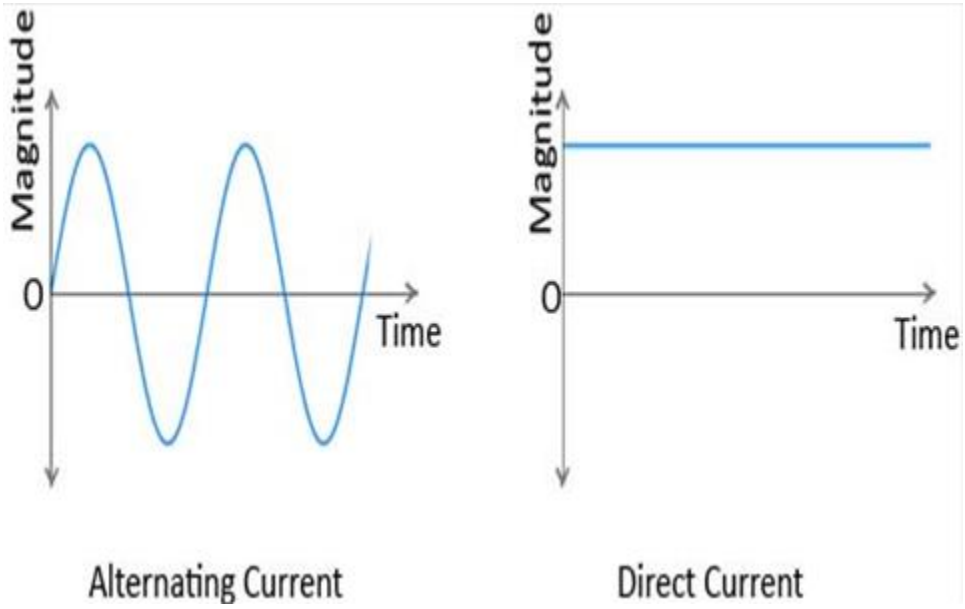
- The CPU and other important computer components fit onto the ***Motherboard***, as shown in the figure on the side.

- A ***Motherboard*** is an electronic circuit board which holds and interconnects the hardware together, mainly the CPU and Memory. It also provides connectors for other peripherals.

- The ***Motherboard*** in a desktop computer is housed in a ***case*** (or ***chassis, or system unit***) which collects main hardware components in one place.

- The computer ***case*** collects in addition to motherboard:
    - Hard Drives
    - Optical Drives
    - Coolers and Fans
        - Dissipate the overheat generated by the electronic components, while keeping them operating at acceptable temperature level.

- Power Supply:
  - Feeds computer hardware with electricity. It also converts Alternative Current (AC) (around 220 Volts) from standard wall outlet to Direct Current (DC) (ranging from 5 to 12 Volts) which is the suitable level of current that computer hardware operates properly.



Activity:

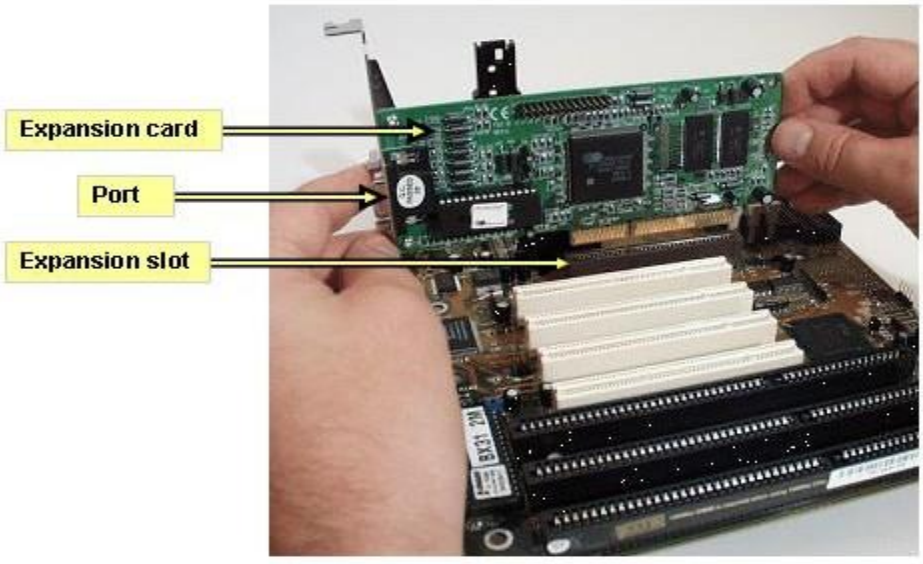
Had you ever heard about UPS? What is it, and for what it is supposed to be used?

- The motherboard contains the ***System Clock***, which is a small quartz crystal circuit to control and synchronize the timing of all computer operations.
- Though, computer processing speed is identified by the frequency of this crystal (in *Hertz*).
- The higher the CPU clock means that more instructions can be processed per second.

Activity:

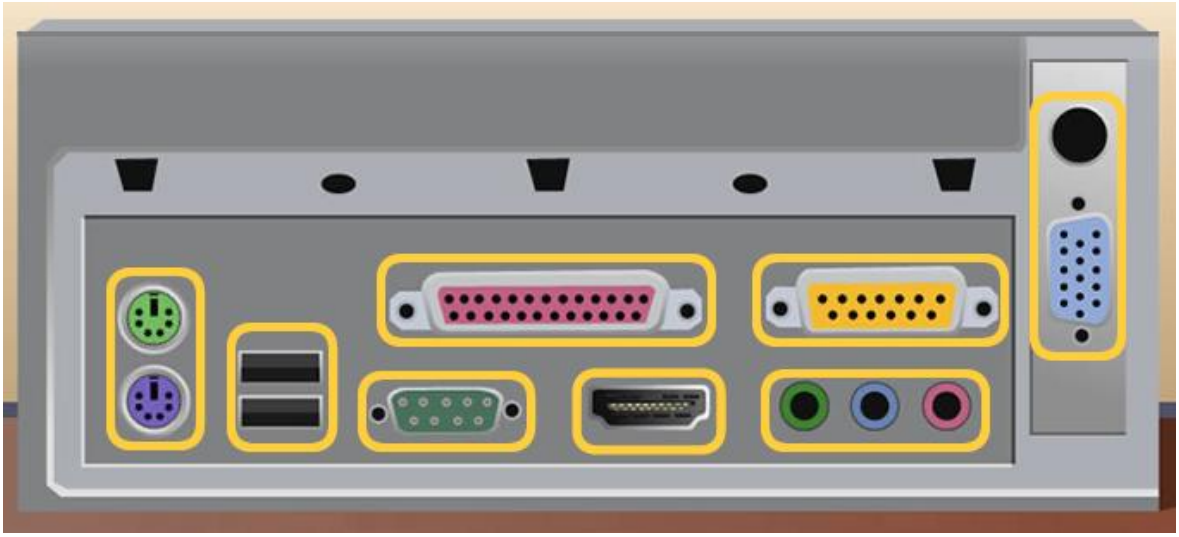
- What's meant by a 2.4 GHz computer processor speed? How the clock speed influences a computer's performance?
- What is a multi-core processor?

- ***Expansion Slots:***
  - ***Expansion slots*** are locations (or sockets) on the motherboard into which expansion cards (also called adapter cards) can be inserted to connect those cards to the motherboard.
  - An ***adapter card*** is a circuit board that gives computers additional capabilities, and expand the computer's functionality, like a video card, network card, or sound card.
  - An ***adapter card*** also provides extra connections to peripheral devices.



- ***Computer Ports***:

The motherboard also contains the Ports that are used to connect the computer to external devices (peripherals).



- PS/2 port - Used to connect a mouse and a keyboard.

- USB Port (universal serial bus) - Designed to be universally compatible with a wide variety of devices.
- Parallel port - Used to connect an external device, such as a printer.
- Serial port - Remote control of equipment, such as routers, switches and mouse.
- HDMI port - Used to connect the video monitor or multimedia projector.
- Game Port - Used to connect joysticks.
- Audio input and output - Used to connect speakers and microphones.
- Video Port - Used to connect a video monitor or multimedia projector.

Activity:

What is Type-C USB port? How it differs from its predecessor USB port?

MEMORY VS. STORAGE

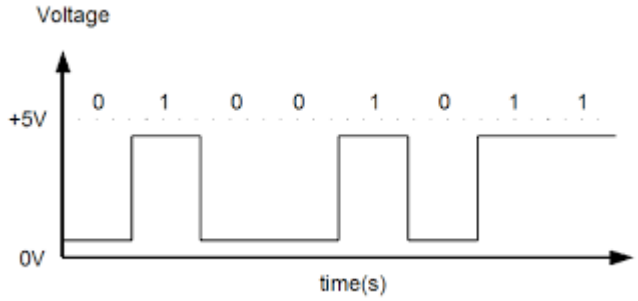
- ***Memory*** refers to locations, usually inside the system unit (typically random access memory or RAM) that a computer uses to store data on a temporary basis. Typically, ***Memory*** refers to chip-based storage. ***Memory*** usually consists of one or more chips on the motherboard or some other circuit board in the computer.

- S***torage*** refers to the more permanent storage a computer uses usually in the form of the computer's internal hard drive or removable storage media (such as DVDs and flash memory storage systems), but it can also be in the form of chip-based internal storage—especially in mobile devices.

MEMORY CAPACITY

- A ***Byte*** is the basic storage unit in memory.
- When running an application, data and instructions are ***loaded*** from storage devices into memory.
- Instructions and data exist as bytes in memory.

- A computer is a ***digital device*** where the internal physical and electronic construction (mainly the transistor which is used as an ON/OFF switch) leads to handle data and instructions as patterns of ZEROs and ONEs. The figure below shows a sample digital signal inside a computer.



- Integrated circuits ("chips") (IC), are the building blocks of computer hardware, are comprised of millions or even billions of transistors.

- The 0's and 1's are typically called ***bits*** (**Bi**nary Digi**t**).

- A ***Byte*** is a group of bits operated on as a single unit in a computer system, which consists of eight bits.

- The following table shows the measurements of memory sizes (capacity):

1 bit	0 or 1
1 Byte	= 8 bits = 2^3 bits
1 KiloByte (KB)	= 1024 Bytes = 2^{10} Bytes = $2^{10} * 2^3$ bits = 2^{13} bits
1 MegaByte (MB)	= 1024 KB = 2^{10} KB = $2^{10} * 2^{10}$ Bytes = $2^{10} * 2^{10} * 2^3$ bits = 2^{23} bits
1 GigaByte (GB)	= 1024 MB = 2^{10} MB = $2^{10} * 2^{10}$ KB = $2^{10} * 2^{10} * 2^{10}$ Bytes = 2^{33} bits
1 TeraByte (TB)	= 1024 GB = 2^{10} GB
1 PetaByte (PB)	= 1024 TB = 2^{10} TB

Activity:

A memory chip has a 32 KB capacity, answer the followings:

a) How many bits can be stored in this memory?
b) What is the capacity of this memory in GB?

VOLATILE VS. NON-VOLATILE MEMORY

- ***Volatile Memory:*** Loses its contents when the power is turned off.
- ***Non-Volatile Memory:*** Keeps its contents when the power is turned off.

TYPES OF MEMORY

[1] Registers (Volatile memory inside the processor***)***

[2] RAM (Random Access Memory):

- RAM is part of the *main memory*.
- It is Volatile.
- Used to store the essential parts of the operating system while the computer is running, as well as the programs and data that the computer is currently using.

Activity:

- What is the RAM size of your computer?

- Can you increase the RAM size in your mobile device?

- How RAM size affects the computer performance?

- Why when we open a program for the first time it takes longer time to start than the second time?

[3] ROM (Read-Only Memory):

- Non-Volatile memory chips attached to the motherboard.
- Contents of ROM cannot be changed or updated.
- Mainly, ROM was used to store data and programs responsible for starting up the computer.

Activity:

What are the main differences between RAM and ROM in terms of capacity, volatility, and uses?

[4] Flash Memory:

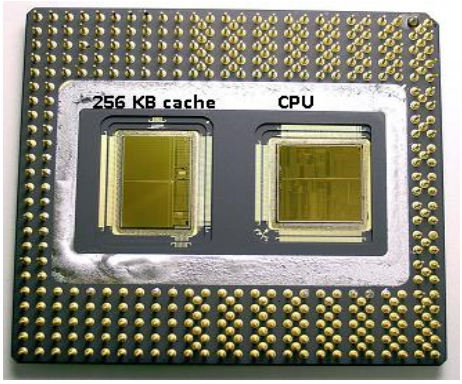
- Non-Volatile memory chips.
- Replaced the ROM chips when storing system information required during the booting (starting up).
- Unlike ROM, the contents of Flash Memory may be updated and changed.
- Flash memory chips are also built into many types of devices (such as tablets, handheld gaming devices, and smartphones), as well as built into storage media and devices (such as flash memory cards and USB flash drives).

Activity:

What is meant by the term POST (***power on self-test***), and what is meant by this term?

[5] Cache Memory:

- Volatile memory chips.
- ***Cache memory*** is used to speed up processing by storing the data and instructions that may be needed next by the CPU in handy locations.
- Very fast memory
- Very expensive
- Small capacity (few MBytes)
- Usually comes as part of the CPU



Activity:

As the Cache memory is faster than the RAM, why do not we replace the RAM with Cache?

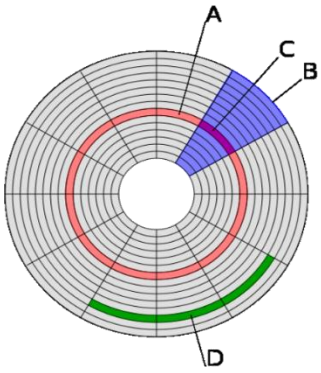
TYPES OF STORAGE

- Storage may also be known as ***Secondary Storage***, or ***Auxiliary Storage***.
- A Storage Device is usually used to store data and programs permanently (Non-Volatile).
- We need storage devices for information to be recorded. A computer has devices that store information so that it is not deleted when the computer is turned off.

- Data stored on a storage device using any of the following techniques:
  - Magnetically
  - Optically
  - Flash memory

- We'll study The following storage technologies:
  - Hard Disk Drive
  - Solid State Disk
  - Optical Disk
  - Magnetic Tape
  - Cloud Storage

[1] Hard Disk Drive (HDD)



Hard Drive Structure:

A = track
B = sector
C = sector of a track
D = cluster

- Uses magnetic field to store data.
- Electro-mechanical device.
- Data is organized as tracks and sectors.
- Huge capacity, ranging from GigaBytes to few TeraBytes.
- Not expensive.
- Relatively Slow.

[2] Solid State Disk (SSD)

- Uses flash memory technology to store data.
- New Technology which replaced HDD in some new computers.
- Big capacity (100 GB – 16 TB).
- Fast; faster than Hard Disk.
- Relatively expensive.
- SD memory card and USB flash drive also use flash technology to store data.

[3] Optical Disk

- Slow.
- Not expensive.
- Capacity range from 700 MB – 100 GB
- Use Laser Beam to write/read data from disk, this includes CD, DVD, and Blu-ray disks.

Types of Optical Drives:

- CD (Compact Disk)
  - Size (up to 700 MB)
  - Cheap
  - All computers with CD drive can read it
- DVD (Digital Versatile Video)
  - Size (1.8 GB – 8.5 GB)
  - Originally used to store movies
  - Only optical drives with DVD label can read it
- Blu-ray
  - Capacity (25GB – 100GB)
  - Developed by Sony
  - Requires special Blue-ray drive to read
  - Used to store 3D Movies and Big Video games

Activity:

What is meant by each of the followings?

- Optical Drive
- Optical Disk

[4] Magnetic Tape

- Very slow (Sequential Access)
- Cheap
- Good capacity (10 GBytes – 100 GBytes)
- Usually used for long-term data archiving and back-up.

Activity:

Why do you think Magnetic Tapes are suitable for back-up storage?

[5] Cloud Storage

- **Cloud storage** is a service that allows you to save and access your files over the internet, rather than storing them on your local device or physical storage.
- Examples: Dropbox, Google Drive, iCloud, and Microsoft OneDrive.
- Instead of storing data locally, store it off-site on the Internet (Hosted by servers).
- Advantages include: Cost, Accessibility, sharing, security.
- Disadvantages include: privacy, requires Internet connection.

Discussion:

- How the use of Cloud Storage would make your data more accessible?

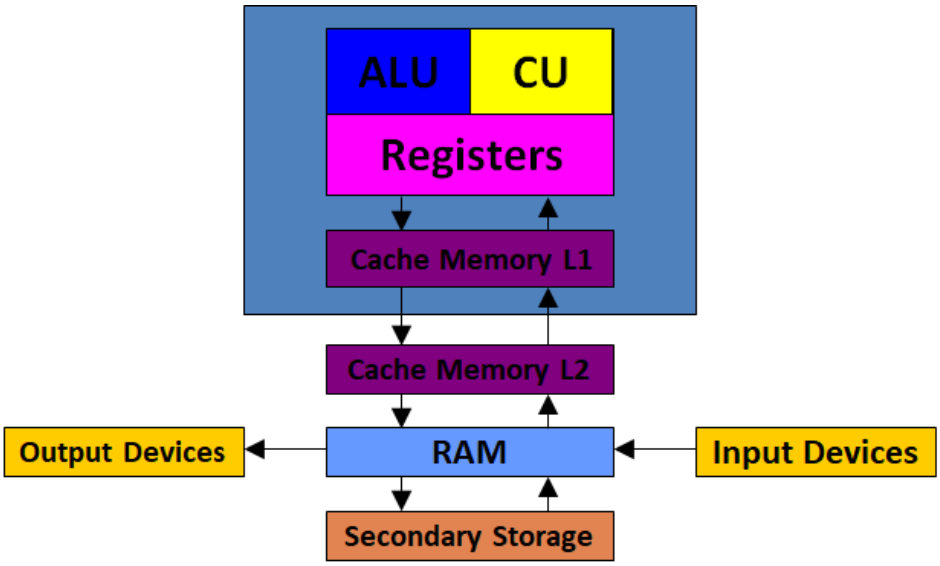
- Using the Cloud for storage raises some privacy concerns, explain?

MEMORY HIERARCHY

Why do we need all these types of memory/storage devices?

Because we need to balance between these three factors:

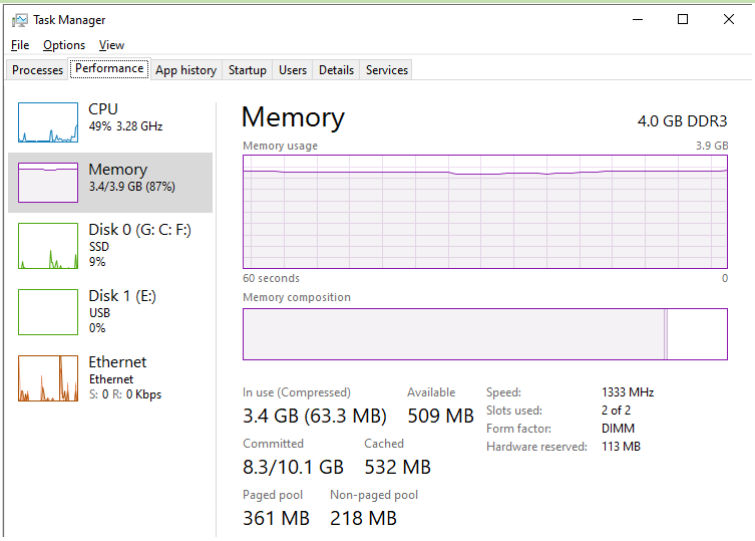
- Cost
- Speed
- Capacity



COMPUTER PERFORMANCE

Depends on many factors:

- CPU Speed (clock rate).
- Number of CPU Cores.
- Cache and RAM sizes .
- Secondary Storage speed; HDD or SSD.



- Input Components:
  - Used to enter data and commands into a computer like Keyboard, Scanner, Microphone, Mouse, Webcam, Touch Screen, …

- ***Keyboard***: A device that has a series of buttons with letters, numbers and symbols, used to enter data.
- ***Webcam:*** Video camera that captures either still pictures or motion video, and can transmit its video on the Internet in real-time. These days, it's commonly used for videoconferencing.

Digital Cameras and WebCams

- ***Mouse***: A pointing device that controls the cursor around the monitor.

Touchpad

Trackball

- ***Microphone***: Captures audio and human voice.
- ***Magnetic strip reader***

Joystick

Microphone

Magnetic-strip reader

- ***Barcode reader***
- ***Game Controllers:*** designed for gaming, including gamepads, joysticks, and steering wheels.
- ***Scanner***: Scans images, photos and text.

Scanner flat-fed Scanner sheet-fed

- Scanner flat-fed
- Scanner barcode-reader
- Scanner hand-held

- Output Components:
  - Responsible for providing the user with information from computer output (images, graphics, sounds, audio, videos, text and others …) like Screen, Printer, Speaker, Projector, …

- ***Monitor***: Displays system information visually.
- ***Printer:*** Prints text, graphics and images on a paper.
- ***Speaker***: Propagates the sound and improves sound reproduction.

MONITORS

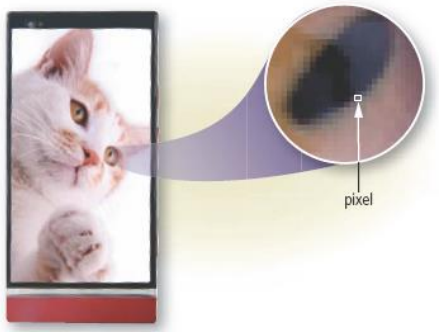
- A Monitor (or Screen, or Display device) is an output device that produces visual information to the user. The displayed information called ***soft copy*** information.
- Display Technologies:
  - CRT (Cathode-Ray Tube)
  - LCD (Liquid Crystal Display)
  - LED (Light-Emitting Diode)

Monitor (Display) CRT

Monitor (Display) LCD

LED Display

- The basic element of an image being displayed is called a **Pixel** (short for Picture Element).

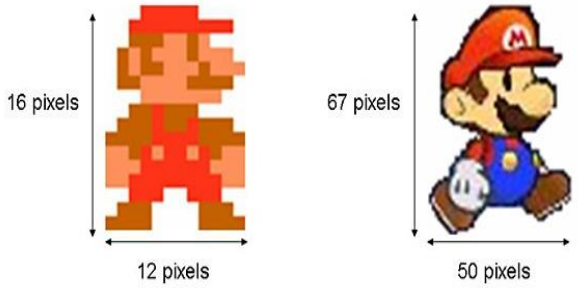


Display Features:

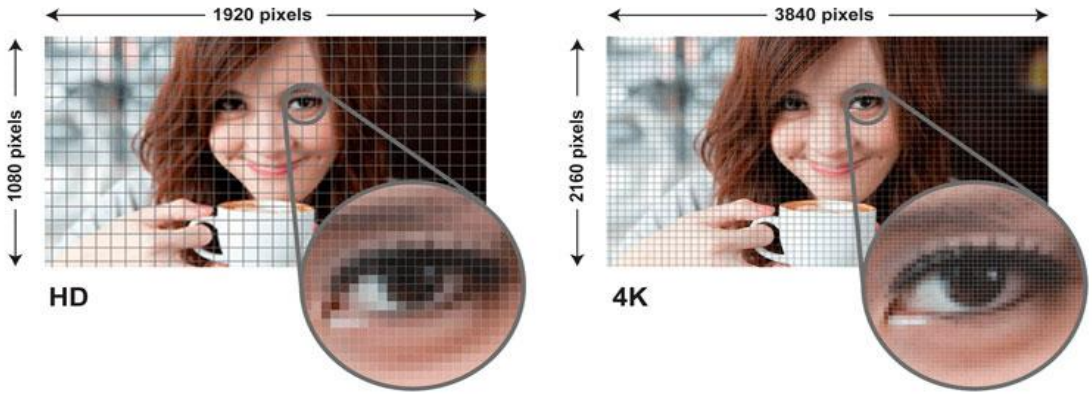
[1] Resolution

- ***Resolution*** is the number of horizontal and vertical pixels in a display (rows × columns). For example, a monitor or screen that has a 1600×900 resolution displays up to 1600 pixels per horizontal row and 900 pixels per vertical row, for a total of 1,440,000 pixels to create a screen image.

- The higher a monitor's resolution (the more pixels), the clearer the image produced.



- PPI stands for Pixels per Inch and is a metric typically used to describe the pixel density (sharpness) for all sorts of displays, including cameras, computers, mobile devices, etc…



[2] Dot (Pixel) Pitch

- ***Dot pitch*** is the distance in millimeters between pixels on a display. Text created with a smaller dot pitch is easier to read.
- The lower the number, the sharper the image.

[3] Size

- The display size is measured by the diagonal length of a monitor's viewing area.
- Common sizes are 15", 17", 19", 21", and 24" inches.

[4] Aspect Ratio

- The aspect ratio defines a display's width relative to its height.
- 2:1 aspect ratio, for example, means the display is twice as wide as it is tall. The aspect ratio for a widescreen monitor is 16:9 or 16:10.

- 4:3 means that the ratio of the width of the display screen to the height is 4 to 3.

Other display devices:

1) Projectors
2) Interactive Whiteboard
3) HDTV (high-definition Television)

Digital projector

PRINTERS

- A printer is an output device that translates information on paper.
- Printer output is called ***hard copy*** information.

Printer Features:

[1] Resolution

- ***Resolution:*** measured by **dpi** (dot per inch).
- The higher the dpi, the better the quality of images produced.
- Most printers designed for personal use average 1,200 by 4,800 dpi.

[2] Speed

- ***Speed:*** measured by the number of pages printed per minute (ppm).

[3] Color

- ***Black/White***.
- ***Colored.***

Printer Types:

- Inkjet Printers
  - Spraying tiny drops of liquid ink onto a piece of paper.
  - Works with ink cartridges to print text, spreadsheets, photos and graphics.

- Laser Printers
  - High-speed, high-quality printer.
  - Creates images using a laser beam and powdered ink, called toner.

- Thermal Printers
  - Used by ATMs, Cashiers, point-of-sale (POS) to print receipts.
  - Small and Light-weight.
  - Use special heat-sensitive paper.

- Plotters
  - Plotters are sophisticated printers used to produce high-quality large-scale drawings, such as banners, blueprints, maps, and circuit diagrams.

- All-in-One Printers
  - Also called a multifunction printer.
  - Integrates printing, copying and scanning functions into one machine.
  - It may also support fax service.
  - Some use color ink-jet printer technology, while others use laser technology.

Activity:

You are kindly requested to refer to the Internet to know more about a trendy printer called: **3D Printer**.

Activity:

- What is a QR code?

- Scan the aside QR code by your mobile and see what you will get!

CHAPTER QUESTIONS:

Q1) State whether each of the following statements is **TRUE** or **FALSE**.

	Statement	TRUE / FALSE
[1]	The keyboard, mouse, microphone, scanner, digital camera, and PC camera are six commonly used output devices.	
[2]	During processing, the processor places instructions to be executed and data needed by these instructions into memory.	
[3]	Storage differs from memory in that it holds items only temporarily while the processor interprets and executes them.	
[4]	A storage medium is the physical material on which a computer keeps data, instructions, and information.	
[5]	When a computer is started, the operating system loads onto the computer's hard disk from memory.	
[6]	On a personal computer, the electronic components and most storage devices reside outside the system unit.	
[7]	A hertz is one cycle per second.	
[8]	A CPU with higher clock speed can process fewer instructions per second than a CPU with a lower clock speed.	
[9]	The higher the clock speed, the faster the processor, and the more expensive the computer.	
[10]	Nonvolatile memory loses its contents when power is removed from the computer.	
[11]	The output that produced from printers is called soft copy.	
[12]	Optical drive is the physical medium where data is stored optically.	
[13]	For an image with total of 128 pixels and each pixel is 8 bit, then its final size is 1 KB.	
[14]	All programs that currently executing must be loaded into the ROM.	
[15]	A Solid State Disk (SSD) is faster than Hard Disk Drive (HDD).	

Q2) Choose the best answer:

1	Computer hardware includes all of the following *except* _____.	**A.** input and output devices **B.** storage and communications devices **C.** a system unit **D.** application programs
2	Some computer components, such as the _____, are internal and reside inside the system unit.	**A.** keyboard and mouse **B.** monitor and microphone **C.** processor and memory **D.** printer and scanner
3	The CPU contains _____.	**A.** input devices and output devices **B.** the control unit and the arithmetic/logic unit **C.** main memory and storage devices **D.** all of the above
4	Arithmetic operations _____.	**A.** involve matching one data item to another to determine if the first item is greater than, equal to, or less than the other item **B.** include addition, subtraction, multiplication, and division **C.** sort data items according to standard, predefined criteria in ascending order or descending order **D.** use conditions with operators such as AND, OR, and NOT
5	The data stored on ROM chips _____ when power to the computer is turned off.	**A.** can be modified and is lost **B.** can be modified and is not lost **C.** cannot be modified and is lost **D.** cannot be modified and is not lost

6	Which of the followings is a permanent memory type?	**A.** RAM **B.** flash **C.** Cache **D.** Registers
7	2^{10} GB is equal to:	**A.** 1 TB. **B.** 1024 GB. **C.** 2^{20} MB. **D.** All of the above.
8	The memory used to store data and programs responsible for starting up the computer is:	**A.** RAM **B.** ROM **C.** Flash **D.** CD
9	The part of CPU that is responsible for controlling operations and communications:	**A.** ALU **B.** CU **C.** Registers **D.** Cache
10	One of the following printers uses spraying to print information:	**A.** Inkjet Printer **B.** Laser printer **C.** Thermal Printer **D.** Dot Matrix printer
11	All of the following are advantages of cloud storage except:	**A.** Data Sharing. **B.** Security. **C.** Privacy. **D.** Accessibility
12	The computer performance is affected by many factors like:	**A.** RAM capacity **B.** Processor speed **C.** Cache size **D.** All of the choices
13	The CPU speed is measured in:	**A.** Mpbs **B.** Giga Bytes **C.** GHz **D.** ppm

Q3) An image is presented on a screen containing 360 pixels width, and 180 pixels height, answer the followings:

A) What is the total pixels being displayed on the screen?

B) What is the aspect ratio that best views this image?

C) If each pixel is represented as 8 bits,

- What is the image size in KB?
- What is the image size in MB?

جامعة بوليتكنك فلسطين

COMPUTER SOFTWARE

Prepared by:
Eng. Yousef Salah
Eng. Mohammad Jabari

This material developed under the objectives of FESTEM project funded by the EU.
https://festem.ps/

Co-funded by the Erasmus+ Programme of the European Union

CHAPTER THREE
COMPUTER SOFTWARE

PREFACE:

This chapter focuses on computer software and how programs enable computers to function efficiently for users. It distinguishes between the two main types of software: system software and application software, providing detailed examples of each. The chapter also explores the operating system as a crucial component of system software, highlighting its core functions. Additionally, it discusses utility programs that handle maintenance tasks for computers.

INTENDED LEARNING OUTCOMES:

After completing this chapter students will be able to:

1) Describe the differences between system software and application software.
2) Understand what application software is, and provide examples about it.
3) Explain the basic functions, features, and categories of operating systems.
4) Name today's most widely used operating systems.
5) Explain the purpose of utilities and utility suites.

FURTHER READING:

1) Discovering Computers ©2018: Digital Technology, Data, and Devices.
2) Computing Essentials 2017-McGraw-Hill (2017) Daniel O'Leary, Linda I. O'Leary, Timothy J O'Leary.
3) Understanding Computers Today And Tomorrow Comprehensive, Deborah Morley, Charles S. Parker - Cengage Learning, (2016).

WHAT IS COMPUTER SOFTWARE

- A ***computer software*** (or program, application) is the list of instructions that tell the computer what to do, and how to process data to get what the user wants.
- Computer software drives the hardware to operate properly.
- Types of Software:
  - Application Software
  - System Software

Application Software:

- Performs a specific task, and provides a service to the end user.
- The user interacts with the application software.
- Examples of application software:
  - Google Chrome: Web browsing.
  - Notepad: Text editing.
  - Media Player: Audio playing.
  - MS Excel: Spreadsheet application.
  - MS PowerPoint: Presentation software.
  - Photoshop: Graphic software.

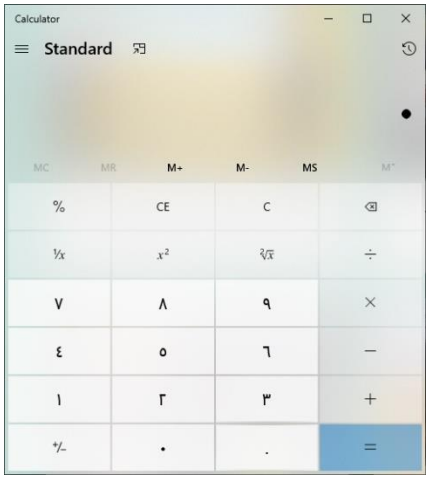
- Application software may be classified into:
  - Desktop Application
  - Web Application
  - Mobile Application

Activity:

List a name of a program for each category of the above application software.

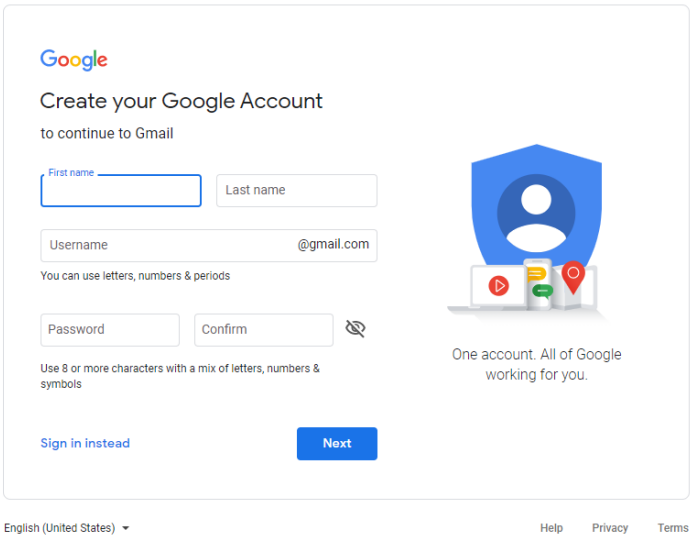
- Desktop Application:

- Runs as a stand-alone program on a PC or laptop.
- It may require an installation process.
- Examples:
  - Calculator, Photoshop, MS office, ….



- Web Application (or web app):

- Runs inside a web browser.
- Examples:
  - Facebook, web-email, Google Classroom, ….



- Mobile Application (or app):

- Runs in a mobile device like smartphone or tablet.
- Examples:
  - Instagram, Google Weather, Telegram, WhatsApp, Google Maps, …

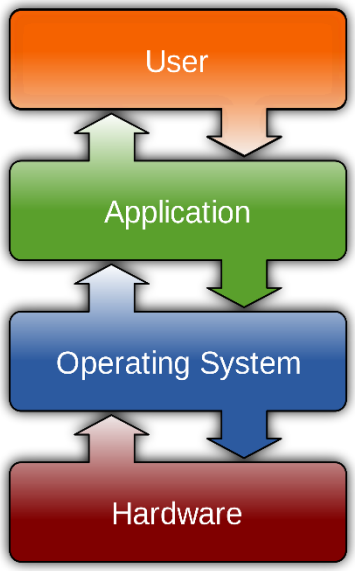


System Software:

- System software are programs that manage and control the operations of computer resources (software, hardware, users, connections, …).
- These programs enable the computer to boot, launch application programs, configure your computer to work with the hardware connected to it, connect your computer to a network, manage files on hard drive, and protect your computer from unauthorized use.
- Examples of system software:
  - Operating Systems
  - Device Drivers
  - Utility programs such as disk defragmentation, back-up, etc..

- Operating System:

- An Operating System (OS) is a system software that installed to coordinate and manage all the operations and activities running on a computer.
- OS also provides an interface between application software and computer hardware.



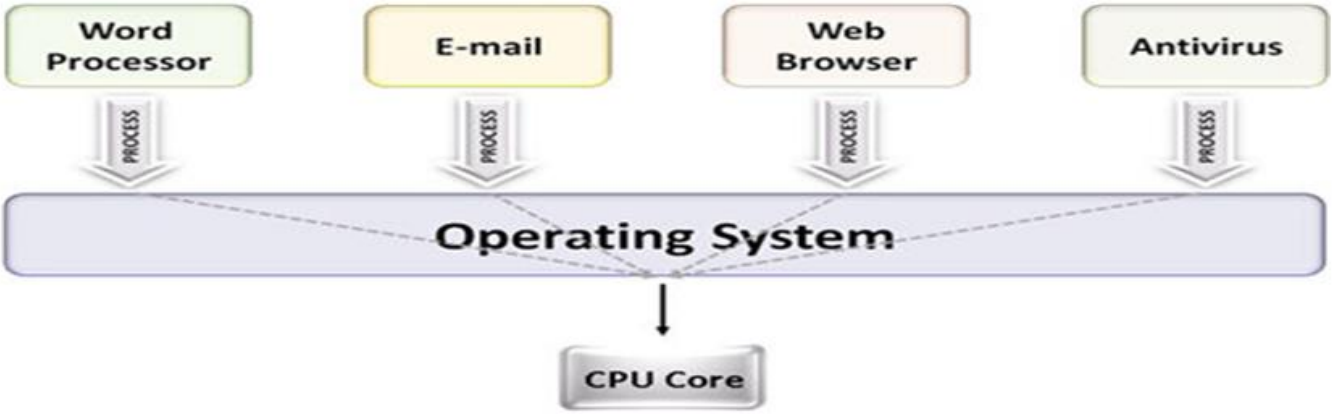
- Examples of Desktop OS:
  - Windows
  - Linux
  - Unix
  - Mac OS

- Examples of Mobile OS:
  - Android
  - iOS
  - HarmonyOS
  - Windows Phone

Main functions of an Operating System:

- **Booting the computer**
  - The OS is the first program that is loaded to run when the computer is turned on.

- **Process Management:**
  - The OS responsible for running computer programs.
  - It also coordinates the execution of multiple programs simultaneously (***Multitasking***).



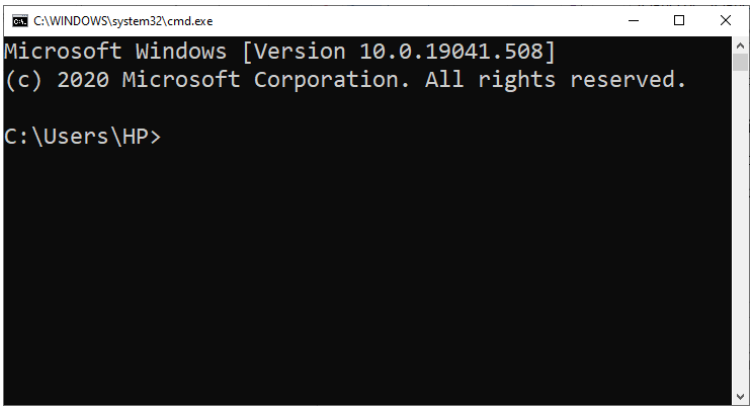
- **Memory Management:**
  - OS loads program into memory for execution in efficient way.
  - The memory is a shared resource, so the OS optimizes the RAM to speed up the processing of programs.

- **Administering Security:**
  - OS uses passwords and biometric characteristics and other security procedures to limit access to the computer and other system resources to only authorized users.
  - OS also manages the privileges and access permissions of multi-user computer systems.
  - Additionally, OS protects the computers from attackers and unauthorized access.

- **User Interface:**
  - OS provides the view at which the user interacts with applications. The interface provides means of inputting data and producing information.

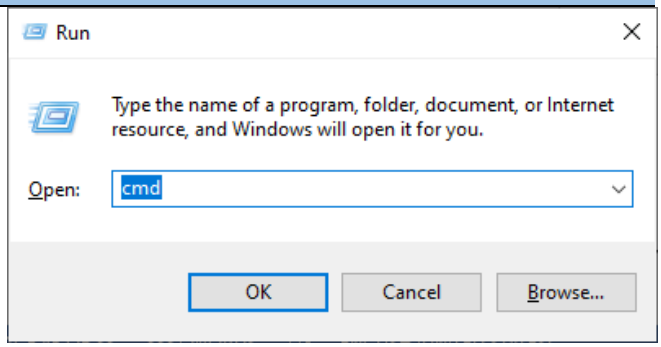
  - User Interface may be:
    - **Graphical User Interface (GUI):**
      The user interacts with visual images by touching, pointing, tapping, or clicking buttons, menus, icons and other objects to issue commands.

    - **Command Line Interface (CLI):**
      The user writes commands by typing text to enter data and instructions.
      (Also called: Terminal, Shell, Console)



Activity:

On Windows® OS, and through the RUN dialogue, write the "cmd" command and view the Windows CLI interface.



- **Configuring Devices:**
  - The OS configures all devices connected to a computer.
  - Small programs called ***device drivers*** (or drivers) are used to communicate with peripheral devices, such as monitors, printers, scanners ….

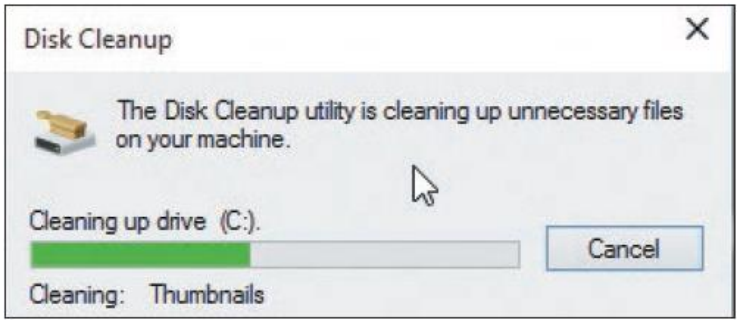
- Nearly, all the operations and services running on a computer is controlled and managed by the OS. For example, the OS is also responsible for File and disk Management, Performance Monitoring, Network and Internet connections, …

Activity:

A single physical computer can support multiple operating systems that operate independently. Search for the term describing this feature.

- **Utility Program:**
  - A utility program is a software usually related to managing or maintaining a computer system. It optimizes the performance of a computer.

  - Examples of Utility programs:
    - Programs for finding files.
    - Cleaning up a hard drive.
    - Backing up software.
    - File Compression.
    - Disk defragmentation.
    - Antivirus Programs.



Back-up Tools:

- Create identical copy of the data (replica).
- Original data can be restored from the back-up.
- Normally, a back-up copy is stored off-site.

Discussion:

- Which desktop OS is most popular today?

- Which Mobile OS is most popular today?

CHAPTER QUESTIONS:

Q1) State whether each of the following statements is **TRUE** or **FALSE**.

	Statement	TRUE / FALSE
[1]	A computer needs an operating system to work.	
[2]	Command-line interfaces often are difficult to use because they require exact spelling, grammar, and punctuation.	
[3]	Smart devices usually have a special operating system.	
[4]	Microsoft Office Word is an example of utility software.	
[5]	A web browser (like Google Chrome) is an example of system software.	
[6]	A device driver is a system software that enables the computer to identify some hardware.	

جامعة بوليتكنك فلسطين

COMPUTER NETWORK AND THE INTERNET

Prepared by:
Eng. Yousef Salah
Eng. Mohammad Jabari

This material developed under the objectives of FESTEM project funded by the EU.
https://festem.ps/

Co-funded by the Erasmus+ Programme of the European Union

CHAPTER FOUR
COMPUTER NETWORKS AND THE INTERNET

PREFACE:

This chapter introduces basic networking principles, including what a computer network is, how it works, and what it can be used for. It emphases on the Internet as a main and daily used network around the world. Moreover, the chapter's purpose is to cover the main terminologies, topologies, and networking architectures. Furthermore, it covers the communication channels, protocols, and networking devices.

INTENDED LEARNING OUTCOMES:

After completing this chapter students will be able to:

1) Define the main terminologies, topologies, and network architectures.
2) Identify the most common communications protocols and networking standards.
3) Describe physical and wireless communication channels.
4) Differentiate among physical transmission media: twisted-pair cable, coaxial cable, and fiber-optic cable.
5) Describe commonly used communications devices.

FURTHER READING:

1) Discovering Computers ©2018: Digital Technology, Data, and Devices.
2) Computing Essentials 2017-McGraw-Hill (2017) Daniel O'Leary, Linda I. O'Leary, Timothy J O'Leary.
3) Understanding Computers Today And Tomorrow Comprehensive, Deborah Morley, Charles S. Parker - Cengage Learning, (2016).

WHAT IS A COMPUTER NETWORK

- A ***computer network*** is a collection of computers and other hardware devices connected together so that network users can share resources (hardware, software, and data).
- Benefits of a computer network:
  - Share hardware.
  - Share data and files, and exchange information.
  - Share applications and software.
  - Support collaboration and interpersonal communication.

THE INTERNET

- The ***Internet*** is a worldwide collection of networks that connects millions of businesses, government agencies, educational institutions, and individuals.
- The Internet is a network of networks.

- After the Internet was founded, many applications and services were widely adopted and used, like:
  - Websites: collection of webpages the user can browse and navigate.
  - Messaging and chatting.
  - Voice calls, video calls, webinars and videoconferencing.
  - E-mail.
  - Forums and blogs.
  - Social networks.
  - E-Commerce: includes online shopping, online auctions, and online payment.
  - Entertainment: Online gaming, TV, movies, music, …
  - Online Education: like Google classroom, Blackboard, and Moodle …
  - Telemedicine.

Classification of Computer Networks:

- Computer Networks can be classified according to the following factors:
  - Geographical span.
  - Topology (Inter-connectivity).

Geographical Span:

A computer network may be classified based on geographical span into:

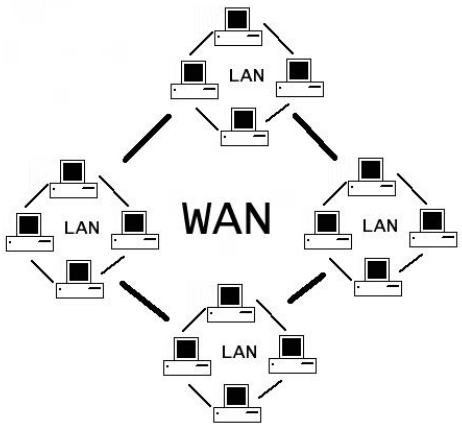
1) LAN (Local Area Network).
2) WAN (Wide Area Network).

LAN (Local Area Network):

- LAN is a computer network that covers small or limited geographical area.
- Example: a network in a home or office, or what spans a floor, building or even two close buildings.

WAN (Wide Area Network):

- WAN is a computer network that extends over large geographical region.
- Example: a network that connects computers and networks across a city or a country.
- Usually, WAN connects multiple LANs together.



Network Topology:

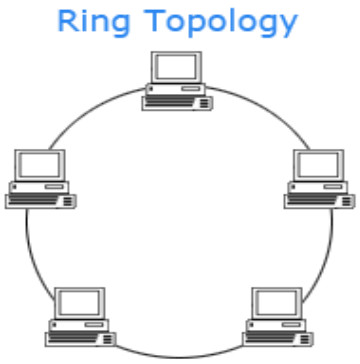
- Network Topology is the physical arrangement in which computer systems or network devices are connected to each other.
- A computer network may be classified based on topology into:
  - Bus Topology.
  - Ring Topology.
  - Star Topology.

Bus Topology

- Uses a central cable to which all network devices connect.
- If the bus line fails, then the network cannot function.

Ring Topology

- Device connections create a circular data path.



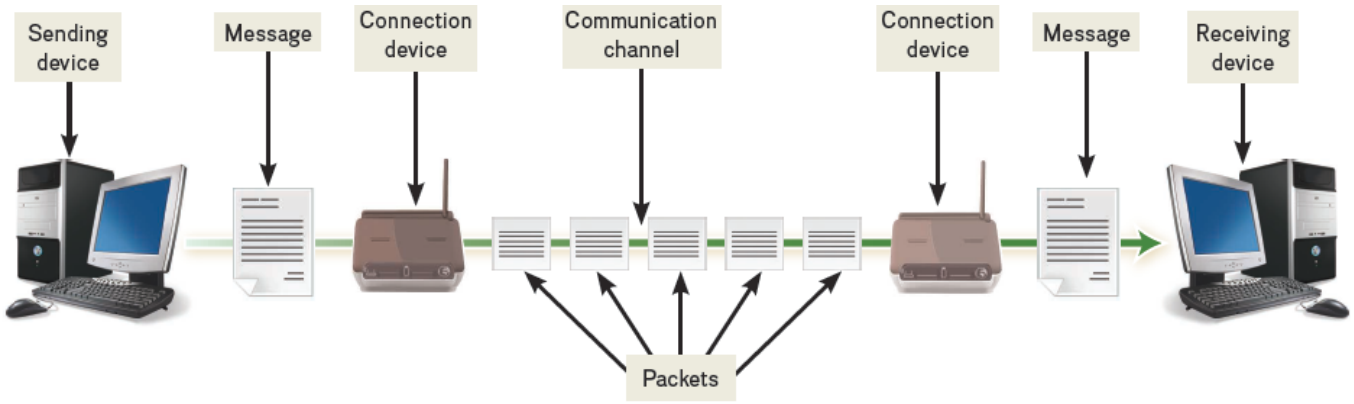
Star Topology

- Uses a central device to which all network devices connect and through which all network data is sent.
- If the central device fails, the network cannot function.

Network Components:

To establish communication between devices on a network we need:

- Protocols.
- Data (messages).
- Transmission Devices.
- Transmission Medium (channel).



Communication Protocols (Standards)

- A ***Protocol*** is a set of rules and procedures that govern correct communication between different parties.
- Internet Protocols (or standards) determine how computers communicate successfully through the Internet
- Examples of Internet protocols:
  - TCP/IP
  - HTTP
  - FTP

Activity:

Write the full text for each of the following abbreviations:

- TCP/IP: _____
- HTTP: _____
- FTP: _____

IP Address

- Each computer on a network is identified by an address called ***IP address***.
- The IP address is a 32 bit address, written as four numbers separated by a dot (or period).
- Each number ranges from 0 to 255.
- IP Example:
    - 212.37.84.129

Activity:

69.514.96.2 is an invalid IP address, why?

Data Packets

- When data is sent over a network connection, the data is decomposed into smaller units called: Packets.
- Each packet has:
    - Sequence number
    - Source address
    - Destination address
    - Data

Transmission Media

- The communication channel at which data travel through the network.
- Communication channel can be:
  - Wired:
    - Twisted Pairs cable
    - Coaxial cable
    - Fiber Optic cable
  - Wireless:
    - IR
    - Bluetooth
    - WiFi and LiFi Technologies
    - Cellular
    - Satellite

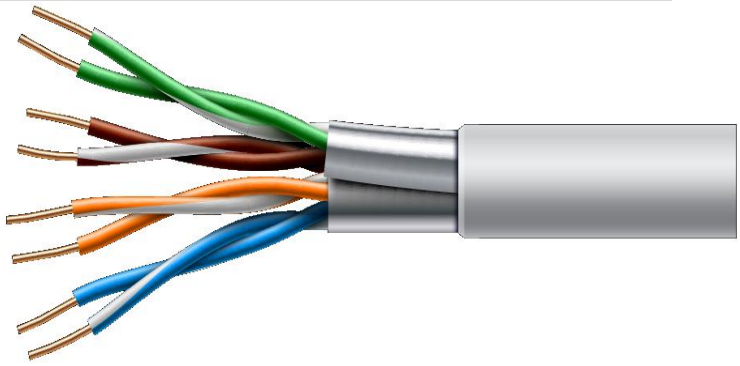
Activity:

Search for the terms:

- Global Positioning System (GPS)?
- Visual Positioning System (VPS)?
- Camera Drones.

Twisted pair cable

- Technology: Copper wires.
- Most common used cable.
- Cheap.
- Speed: 100 Mbps.
- Distance limit: 100 meters.

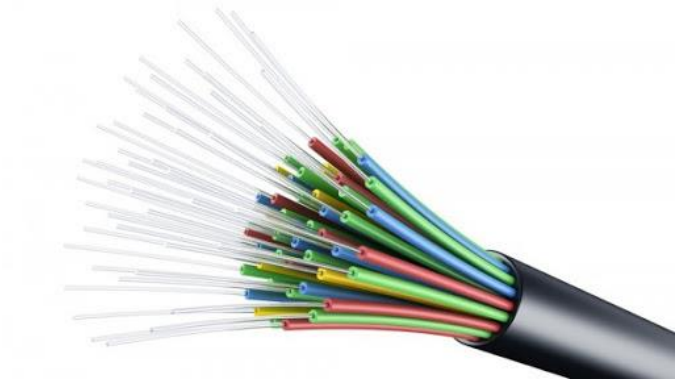


Coaxial cable

- Technology: Copper wires.
- Initially used for TV signal transmission.
- More expensive than twisted pairs.
- Speed: 10 Mbps.
- Distance limit: 500 meters.

Fiber Optics

- Technology: Uses glass threads (fibers) to transmit data as light waves.
- Speed: 10 Gbps.
- Distance: 100 Km.
- Very Expensive.



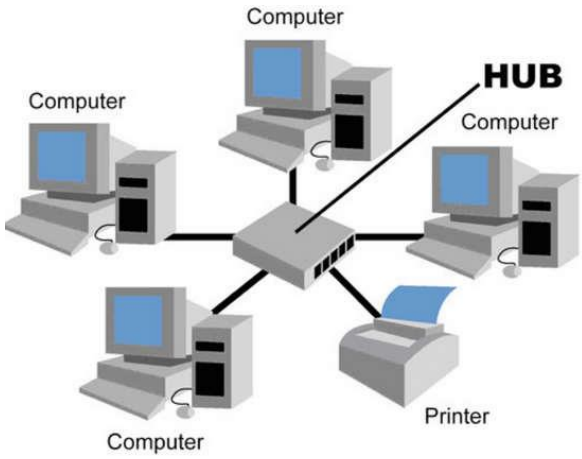
Network Devices

Devices used to create the network, it connects computers and devices together, or connects network with other networks, and most used devices are:

1) Hub
2) Switch
3) Repeater
4) Access Point
5) Router
6) Modem

Hub

- Used in LANs to connect two or more computers.
- Represents a central point.
- It receive messages from one computer and sends it out to all other connected computers.



Switch

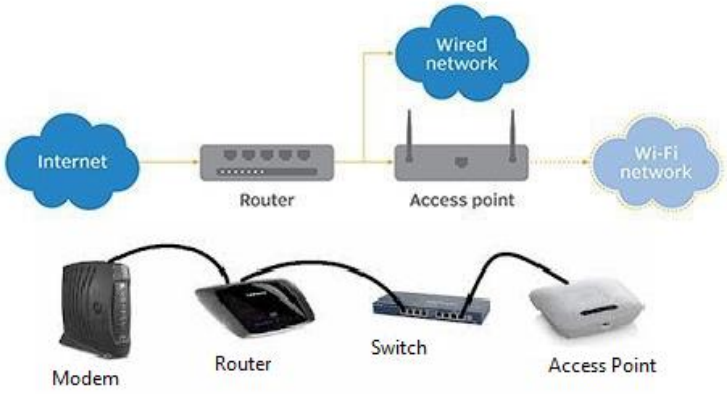
- It is an intelligent Hub.
- Unlike hubs, a switch forwards data packets just to the desired destination.

Repeater

- A **network repeater** is a device that amplifies or regenerates signals in a network to extend the distance they can travel.
- It's commonly used in wired and wireless networks to ensure data can be transmitted over longer distances without signal degradation.
- A repeater boosts the signal to extend the distance it will travel.

Access Point

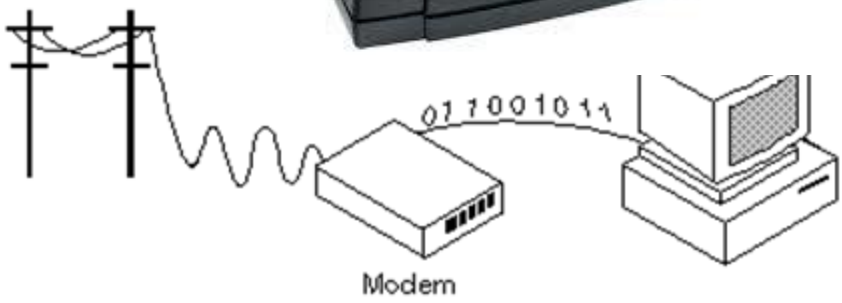
A wireless access point (AP), allows Wireless devices to connect to a wired network.



Modem

- MODulator – DEModulator (Modem)
- Telephone lines transmit data in the form of analog waves, whereas the computer is a digital device.

Analog



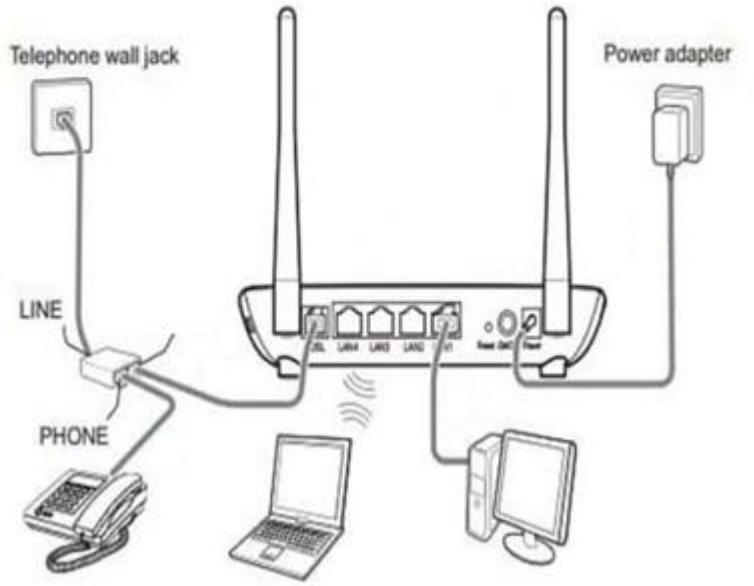
Digital

- The Modem converts signal from analog to digital and vice versa.

Router

- Connects two or more networks together.

ADSL Service



- ADSL: Asymmetric Digital Subscriber Line
- ADSL is a technology used for transmitting digital information on an existing phone line.
- A special filter is installed on a subscriber's telephone line to allow both ADSL and regular voice (telephone) services to be used at the same time.
- The faster technology of ADSL is called VDSL.

Note:

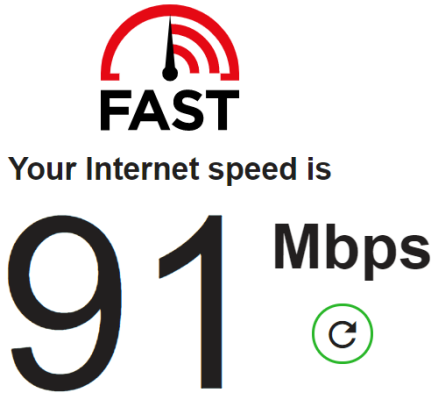
Today, most home ADSL routers combines also an internal modem and a WiFi access point.

ISP (Internet Service Provider):

- ISP is a company that provides the Internet service.

Network Bandwidth

- The amount of information that can be transmitted over a network in a given amount of time, usually expressed in ***bits per second*** (bps).
- ***Downstream*** denotes the speed of downloading data from a network.
- ***Upstream*** denotes the speed of sending data to other devices on a network.



- For the ADSL connections, the upstream is usually 1/8 of the downstream (why?).

Activity:

- Which ISP you are subscribing in your home?

- What is the speed (bandwidth) of this subscription?

- Is it denoting the downstream speed or the upstream speed of the connection?

Assignment:

What is the needed time to download a file with size 1 GB, if you have an Internet connection speed of 8 Mbps?

Discussion:

- Why do we need a modem to connect to the Internet?

- Design a network needed to create an Internet café of 20 computers? What is the needed hardware including computers, cables, peripherals, ….

CHAPTER QUESTIONS:

Q1) State whether each of the following statements is **TRUE** or **FALSE**.

	Statement	TRUE / FALSE
[1]	The Internet is the largest WAN in the world.	
[2]	A cable transmission speed is measured in bps.	
[3]	HTTP is an internet protocol.	
[4]	In the Bus topology, if the shared line failed the whole network will fail.	
[5]	192.186.100.0 is an invalid IP address	
[6]	Twisted pair cable is more expensive than fiber-optics cable.	
[7]	The speed of the upstream is 8 times the speed of the downstream.	
[8]	Switch is an intelligent device, since it can determine the destination of the data packets and forward them appropriately.	
[9]	Coaxial Cable is made of four pairs plastic wires twisted together to send data.	
[10]	Modem is a device that connects two or more networks together.	

Q2) What is the device used in each of the following cases:

a. A device used to connect multiple devices on a wired network. _____

b. A device that enables a computer to communicate over telephone lines. _____

c. A device used to connect wireless devices to a network. _____

d. A device used to amplify signals on a network. _____

جامعة بوليتكنك فلسطين

COMPUTER SECURITY AND PRIVACY

Prepared by:
Eng. Yousef Salah
Eng. Mohammad Jabari

This material developed under the objectives of FESTEM project funded by the EU.
https://festem.ps/

Co-funded by the Erasmus+ Programme of the European Union

CHAPTER FIVE
COMPUTER SECURITY AND PRIVACY

PREFACE:

This chapter provides an overview of computer security, focusing on risks and threats, particularly in the context of online activities. It covers key tools and strategies for protecting computing systems and ensuring authorized access. The chapter also examines malware and cyberattack techniques. Additionally, it emphasizes important topics related to ethics, privacy, and encryption.

INTENDED LEARNING OUTCOMES:

After completing this chapter students will be able to:

1) Provide introduction about computer security, attacks and crimes.
2) List several examples of unauthorized access and unauthorized use and explain several ways to protect against them.
3) Identify the main types of malware and differentiate among these types.
4) Describe some cybercrime ways, and provide examples of online protection systems.
5) Discuss related security topics terminologies like: computer ethics, encryption, mobile apps security and privacy.

FURTHER READING:

1) Discovering Computers ©2018: Digital Technology, Data, and Devices.
2) Computing Essentials 2017-McGraw-Hill (2017) Daniel O'Leary, Linda I. O'Leary, Timothy J O'Leary.
3) Understanding Computers Today And Tomorrow Comprehensive, Deborah Morley, Charles S. Parker - Cengage Learning, (2016).

WHAT IS A COMPUTER SECURITY

- ***Security*** involves protecting individuals and organizations from theft and danger.

- ***Computer security*** focuses on protecting information, hardware, and software from unauthorized use.

- ***Computer crime*** includes any illegal act involving a computer or a computer network.
- C***ybercrime*** is an Internet-based crime.

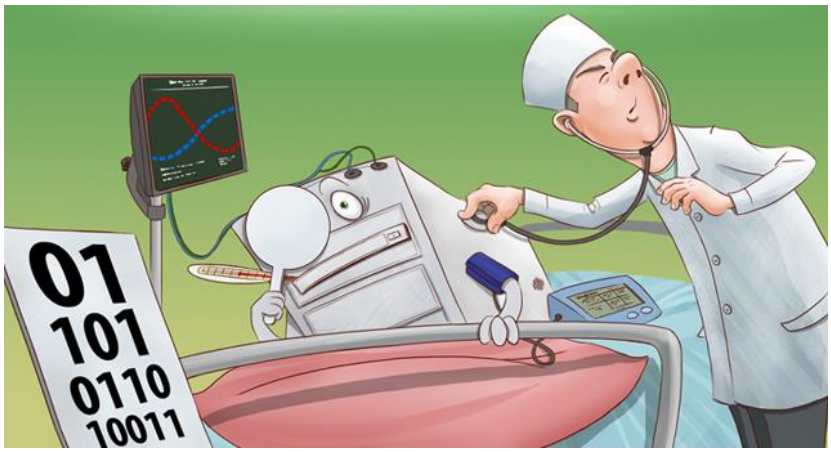
- ***Hackers*** and ***Crackers*** are technically skilled persons who try to get unauthorized access within a computer system or a network.

- Certain computer attacks exploit the vulnerabilities and weaknesses of a computer system to harm it, or perform illegal actions like:
    - Distributing malicious programs.
    - Stealing data and identity theft.
    - Internet scams.
    - Cyberbullying and blackmailing.
    - Sabotage a computer system.

MALWARE

- **Malware** (short for "malicious software") is any software that has malicious intent, and is designed to harm or exploit a computer device or network.
- When a malware executes, it will damage or disrupt a computer system, or will steal private information.

- Some symptoms of malware infection:
  - Your computer is slowing down.
  - Annoying ads are displayed.
  - System Crashes.
  - Pop-up messages.
  - Internet traffic increases.
  - Lack of storage space.
  - Your browser homepage changed.
  - Unusual messages show unexpectedly.



Activity:

Had your system ever infected by a malware? What symptoms appeared to your system?

- The most common types of malware software are:
  - Viruses.
  - Worms.
  - Trojan horses.
  - Spyware.
  - Ransomware.

Viruses

- A virus is a malware program that attaches and replicates itself to other programs or files.
- Computer viruses are often embedded into program or data files (such as software, games, videos, music files and documents).
- When a virus runs, it harms your system, causes damage, affects performance, or even monitors your activities.
- Most viruses comes through an infected removable storage medium, or via an email attachment or a webpage link clicked by the user.

- Antivirus Software:

- A software that detects and removes viruses.
- The antivirus software should be updated continuously to be notified by latest viruses in the world, so that it can discover virus threats when get into your computer.
- Most common antivirus programs are: Kaspersky, Avast, McAfee, Norton 360, Bitdefender ….

Activity:

What other Antivirus software are common? Are they free software?

Trojan Horses

- Do you know the Story of Troy?
- Trojan horse is a malicious program that hides within or looks like a legitimate program. Unlike a virus or worm, a Trojan horse does not replicate itself to other computers or devices.
- Many recent Trojans trick the user by inviting him to a normal ongoing activities (such as the Windows Update service or a warning from a security program or buying a useless program).
- A Trojan horse uses Social Engineering to trick the user.

- What is Social Engineering?
- Social Engineering is a psychological action used by a hacker to deceive the user by offering him an appealing service to get private information, or to catch his acceptance to run a malware program.
- Social Engineering uses emotional aspects to let a user reveals his private information like usernames, passwords, credit card numbers, …..

Activity:

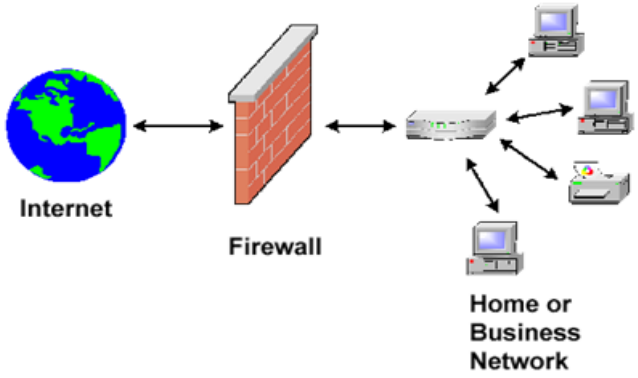
What Social Engineering examples you faced through the Internet?

Ransomware

- Ransomware is a type of malicious software designed to block access to a computer system until a sum of money is paid.

WHAT IS A FIREWALL

- A firewall is a network security system, it protects network resources from intrusions.
- Based on a set of rules, a firewall checks all incoming (from the Internet) and outgoing (to the Internet) traffic, and allows only authorized traffic to pass through the firewall.
- A firewall may be hardware system or software.



SECURITY COUNTERMEASURES

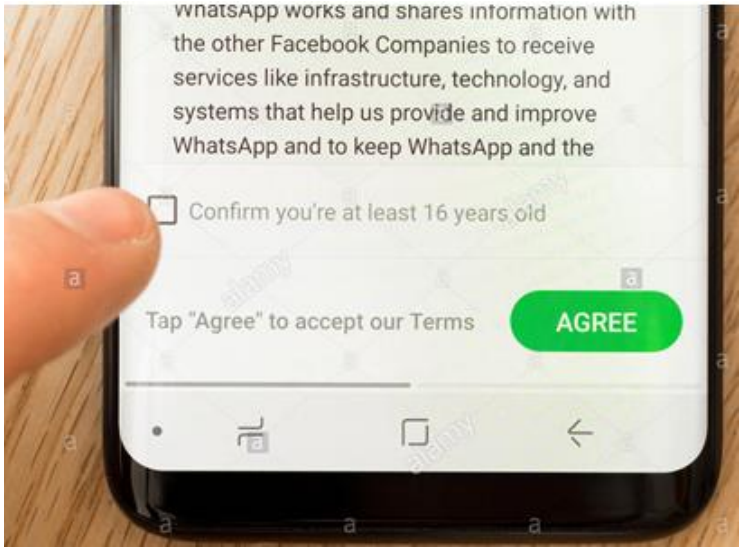
- Countermeasures are preventive procedures and awareness tips that guide the user to be protected and safe against a danger or a threat.
- Main security countermeasures are:
  - Secure your computer: update your operating system and use up-to-date security software (antivirus, antispyware, firewall, etc.).
  - Be suspicious of unsolicited email attachments.
  - Scan removable media for malware before using it.

- Back up your data regularly.
- Use strong passwords, change them regularly, and never give it to others.
- Download files from trusted websites or mobile stores.
- Verify sources before sharing sensitive information—never respond to e-mail or phone requests for sensitive information.
- Avoid putting too many personal details on your web site or a social media site.
- Avoid using location-based services that share your location information with strangers.
- When using a public computer, make sure to delete any personal data or settings before leaving a computer.

PRIVACY

- Privacy is related to what information about individuals is available, how it is used, and by whom.
- **Information privacy** refers to the right of individuals and companies to deny or restrict the collection, use, and dissemination of information about them.

MOBILE APPS SECURITY AND PRIVACY



- The widespread use of mobile devices "apps" raises some serious security and privacy concerns.
- Some Mobile apps may collect data about their users with or without their consent.
- "apps" if given permission, can collect data about your contact list, photos, location, etc..
- It's important to take into account the permission every app asks for before installation.

Activity:

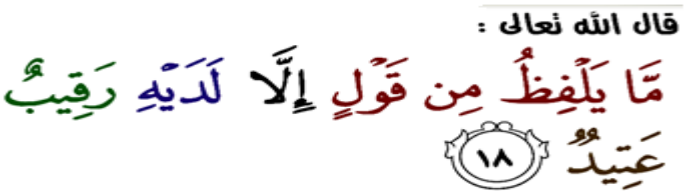
How mobile apps may violate user privacy?

How social media affects privacy?

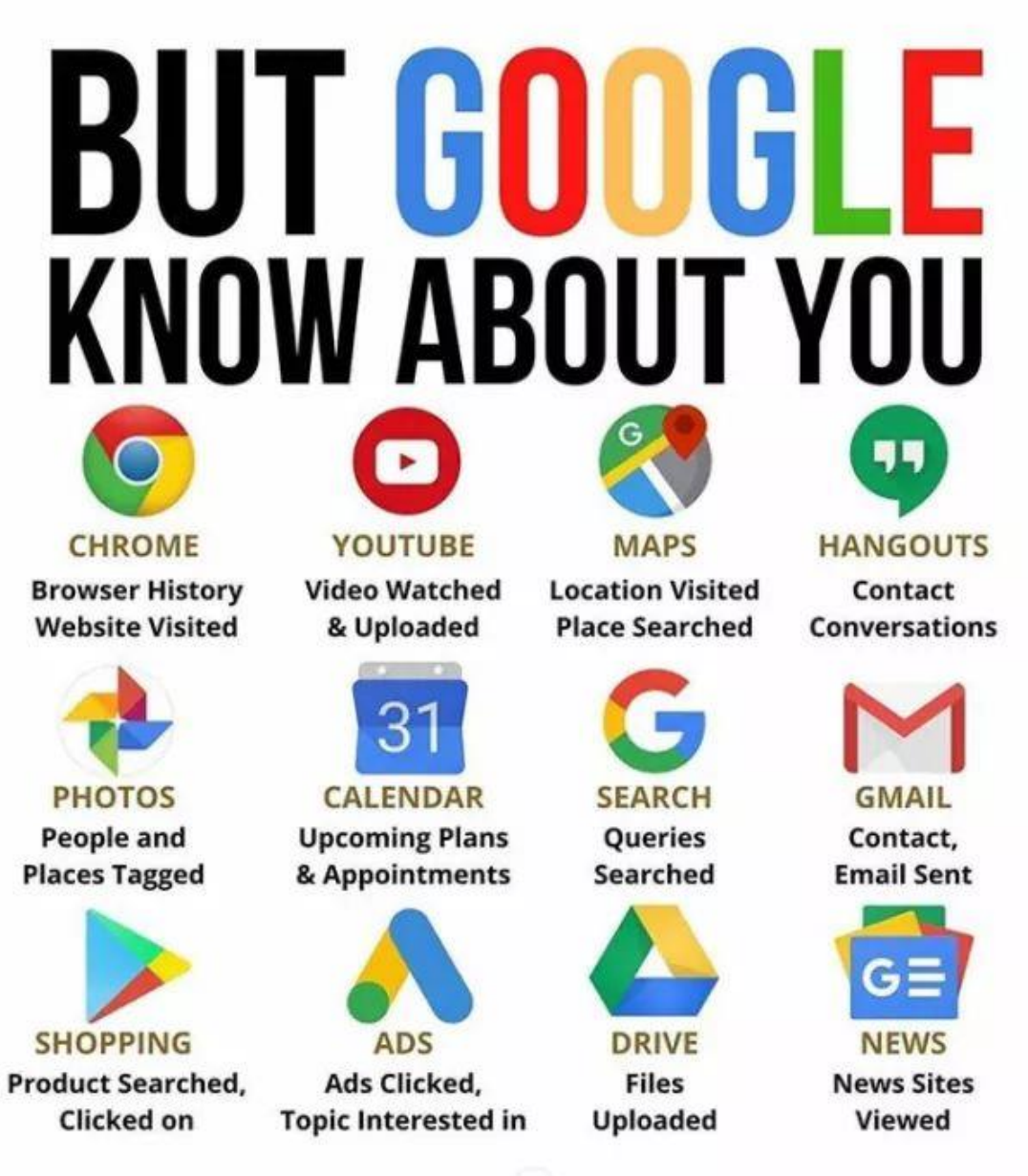
- Social media sites like Facebook, Twitter (now called X), and Instagram, know about you more than you can imagine.
- It knows where you live, whom are your friends, when you plan your next trip, what you like and not like, what you have bought, and much much more!
- You are giving away all these information about yourself for free.

Part of Facebook Privacy Statement

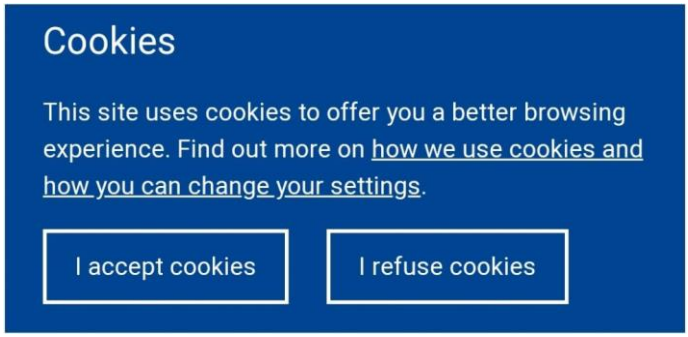
- **Networks and connections.** We collect information about the people, Pages, accounts, hashtags and groups you are connected to and how you interact with them across our Products, such as people you communicate with the most or groups you are part of. We also collect contact information if you choose to upload, sync or import it from a device (such as an address book or call log or SMS log history), which we use for things like helping you and others find people you may know and for the other purposes listed below.
- **Your usage.** We collect information about how you use our Products, such as the types of content you view or engage with; the features you use; the actions you take; the people or accounts you interact with; and the time, frequency and duration of your activities. For example, we log when you're using and have last used our Products, and what posts, videos and other content you view on our Products. We also collect information about how you use features like our camera.



- Be care about what you provide on the Internet.
- You are liable about the content you put while you are online, and you may be consequently prosecuted.
- Even if you used an anonymous way of hiding yourself, then you for sure are religiously questionable.

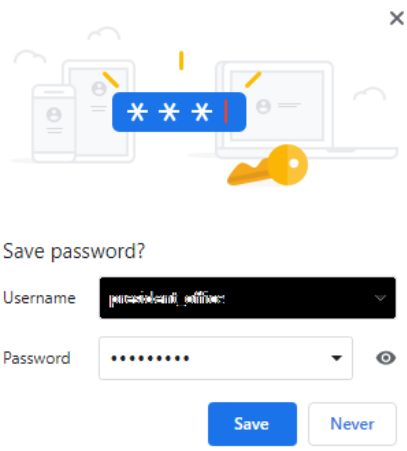


Cookies and Privacy:



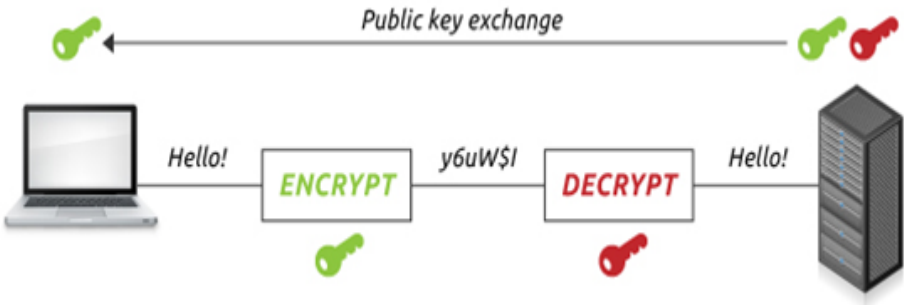
- Cookies are small data files that are saved on your hard disk from websites you have visited.
- Cookies allow these websites to remember you when you visit them next time.
- Based on your browser's settings, these cookies can be accepted or blocked.

- For example, they do not ask you for your "username" and "password" if you check the box "keep me logged in" previously.
- Cookies are useful, but sometimes they are dangerous.
- Some websites use cookies to track users browsing history, behavior and preferences.
- Tracking cookies can be used to display web page ads based on your browsing activities.



- For example if you visited a website that sells Cameras, next time, when you visit Facebook or other websites you will see ads about Cameras, or even your will see ads about the same Camera model you were looking for on that site.

Encryption:



- Data over the Internet, is by default, sent as unencrypted plaintext (*human readable*).
- ***Encryption*** is the process of altering the original representation of information so that it cannot be understood by unauthorized users.
- For example, users may specify that an email application encrypt a message before sending it securely.

Activity:

How **"https"** protocol differs from **"http"** protocol?

COMPUTER ETHICS

- Computer ethics are guidelines for the morally acceptable use of computers in our society.
- Common computer ethics involved in:
  - Intellectual Property rights.
  - Internet and Information Privacy.
  - Code of conduct.
  - Green Computing.
  - Plagiarism.

Sample IT Code of Conduct

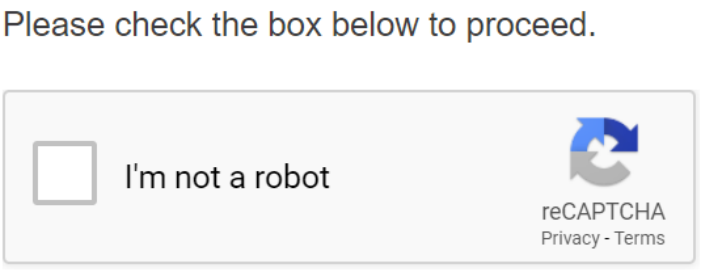
1. Technology may not be used to harm other people.
2. Employees may not meddle in others' files.
3. Employees may use technology only for purposes in which they have been authorized.
4. Technology may not be used to steal.
5. Technology may not be used to bear false witness.
6. Employees may not copy or use software illegally.
7. Employees may not use others' technology resources without authorization.
8. Employees may not use others' intellectual property as their own.
9. Employees shall consider the social impact of programs and systems they design.
10. Employees always should use technology in a way that demonstrates consideration and respect for fellow humans.

Discussion:

- How do companies like Google and Facebook make money?

- How a firewall protects my computer against intruders?

- What is **CAPTCHA**, and what is supposed to achieve?



DISCUSSION: ARTIFICIAL INTELLIGENCE IN EDUCATION

1) Name AI applications and websites that affected the education for both teachers and learners.

2) How AI tools on smart devices facilitate the educational experience?

3) Discuss how AI had an actual impact on the educational ethics.

CHAPTER QUESTIONS:

State whether each of the following statements is **TRUE** or **FALSE**.

	Statement	TRUE / FALSE
[1]	It is safe to check the box "keep me logged in" when using a public computer to check your email.	
[2]	It is impossible to prevent websites from storing cookies on your computer.	
[3]	A Ransomware is a malware program that secretly records and reports an individual's activities on the Internet.	
[4]	It is a good practice to change your password frequently.	
[5]	A cookie is a small text file that a client stores on a web server.	
[6]	A computer virus can only be transferred to another computer via a storage medium.	
[7]	Encryption guarantees altering the shape of information when sent over the network.	
[8]	It is a good practice to avoid putting too many personal details on your website or your social media account.	
[9]	A Firewall provides security through removing viruses from your computer.	
[10]	A Trojan horse malware uses social engineering to replicate itself on other computers.	

جامعة بوليتكنك فلسطين

NUMBERING SYSTEMS

Prepared by:
Eng. Yousef Salah

This material developed under the objectives of FESTEM project funded by the EU.
https://festem.ps/

Co-funded by the Erasmus+ Programme of the European Union

CHAPTER SIX
NUMBERING SYSTEMS

INTRODUCTION

- A number is a value used to represent a quantity that describes a measure or a count.
- A number is expressed as a set of digits and symbols combined together to form a value.

NUMERAL SYSTEM (OR NUMBERING SYSTEM)

- Numeral system is the writing method for expressing numbers.
- It is a mathematical notation for representing numbers of a given set, using digits or other symbols in a consistent manner.
- Each numeral system has a Base (or Radix), which indicates the number of unique numerical digits (or symbols) used to represent numbers.

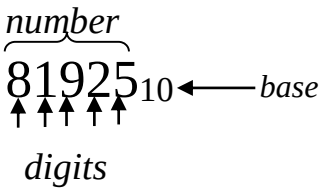
- Examples of numeral systems:
  1- Decimal (Base-10)
  2- Binary (Base-2)
  3- Octal (Base-8)
  4- Hexadecimal (Base-16)

THE DECIMAL SYSTEM

- It is the standard way of representing numbers for humans.
- It is a Base-10 system (Radix = 10) which uses ten digits to form a number: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
- It uses ***positional notation***, where each digit has a position and a weight.

- A number $d_{n-1}d_{n-2} \ldots d_0$ has the digits from position ***zero*** (d_0) to position ***n-1*** (d_{n-1}), where ***n*** is the number of digits in this number.

- Example:



$d_4d_3d_2d_1d_0$

81925_{10}

- In general, if ***b*** is the base, a number in the numeral system of base ***b*** can be expressed in the form:

$$d_{n-1}*b^{n-1} + d_{n-2}*b^{n-2} + \ldots + d_1*b^1 + d_0*b^0$$

- For example:

$$703_{10} = 7\times10^2 + 0\times10^1 + 3\times10^0$$
$$81925_{10} = 8\times10^4 + 1\times10^3 + 9\times10^2 + 2\times10^1 + 5\times10^0$$

- The location of the digit within the number determines the weight b^k.

  For example, the number 27639_{10} (b=10)

most significant digit *least significant digit*

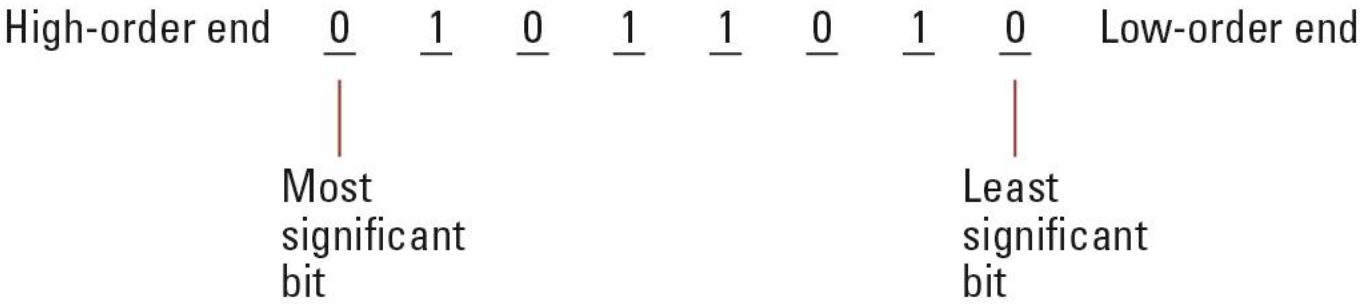
Digit	**2**	**7**	**6**	**3**	**9**
Position	4	3	2	1	0
Weight	10^4	10^3	10^2	10^1	10^0
	10,000	1,000	100	10	1
	2×10^4	7×10^3	6×10^2	3×10^1	9×10^0
	20,000	7,000	600	30	9
Total	= 27639				

THE BINARY SYSTEM

- It is a Base-2 system (Radix = 2) which uses two digits to form a number: {0, 1}
- For example, **10110101_2** is a binary number with $b = 2$.
- The number 1101_2 is equivalent to:
  $1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$
- Each digit in a binary number is called "***bit***" (derived from **Bi**nary Dig**it**).
- The rightmost digit in a binary number is called the *least significant bit* (LSB).
- The leftmost digit in a binary number is called the *most significant bit* (MSB).

Why Binary?

The circuits in a computer are made up of billions of transistors. A transistor is a tiny switch that is activated by the electronic signals it receives. The digits 1 and 0 used in binary reflect the ON and OFF states of a transistor.



Activity: Describe how to count in Binary.

THE OCTAL SYSTEM

- It is a Base-8 system (Radix = 8) which uses eight digits to form a number: {0, 1, 2, 3, 4, 5, 6, 7}
- For example, 6342_8 is an octal number with $b = 8$.
- 2546_8 is equivalent to: $2\times8^3 + 5\times8^2 + 4\times8^1 + 6\times8^0$
- The number that immediately predecessor to 170_8 is: _____
- The number that immediately successor to 507_8 is: _____

Activity: Describe how to count in Octal.

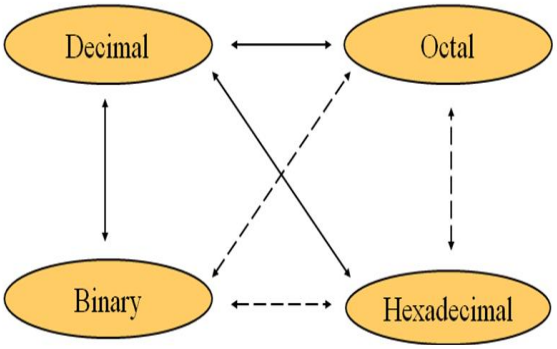
THE HEXADECIMAL SYSTEM

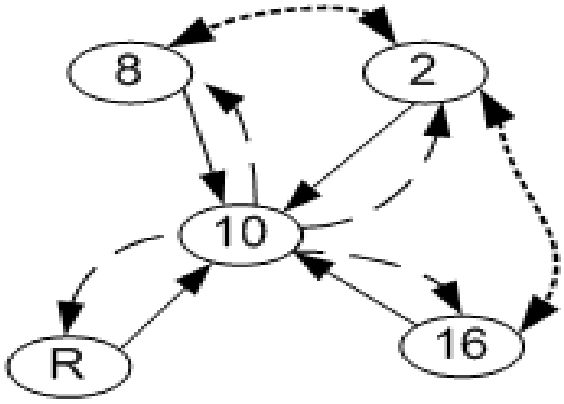
- It is a Base-16 system (Radix = 16) which uses sixteen digits and symbols to form a number: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}, where A=10, B=11, C=12, D=13, E=14, F=15.
- For example, $A62C5_{16}$ is a hexadecimal number with $b = 16$.
- The number $7B2E_{16}$ is equivalent to: $7\times16^3 + 11\times16^2 + 2\times16^1 + 14\times16^0$
- The number that immediately predecessor to $29D0_{16}$ is: _____
- The number that immediately successor to $A5F_{16}$ is: _____

Activity: Describe how to count in Hexadecimal.

NUMERAL SYSTEM CONVERSIONS

- We'll study how to convert from every numeral system to every other numeral system.
- So we'll study 12 conversions:





1 FROM DECIMAL TO BINARY

- Continuously divide the decimal number by ***two*** while keeping track of the remainders, until reaching the result ZERO.
- First remainder is the LSB (least-significant bit).
- Last remainder is the MSB (most-significant bit).

- Example:

$125_{10} = ?_2$

2 | 125		
2 | 62	1	
2 | 31	0	
2 | 15	1	
2 | 7	1	
2 | 3	1	
2 | 1	1	
0	1	

$125_{10} = 1111101_2$

- Exercises:

  What is the binary equivalent value of the following decimal numbers?

  1- 35_{10}

  2- 156_{10}

  3- 4215_{10}

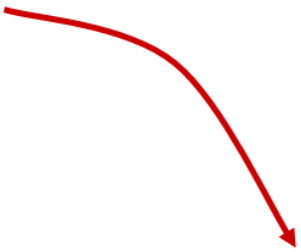
2 FROM DECIMAL TO OCTAL

- Continuously divide the decimal number by ***eight*** while keeping track of the remainders, until reaching the result ZERO.
- First remainder is the LSD (least-significant digit).
- Last remainder is the MSD (most-significant digit).

- Example:

$1234_{10} = ?_8$

8	1234	
8	154	2
8	19	2
8	2	3
	0	2



$1234_{10} = 2322_8$

- Exercises:

  What is the octal equivalent value of the following decimal numbers?

  1- 85_{10}

  2- 214_{10}

  3- 2980_{10}

3 FROM DECIMAL TO HEXADECIMAL

- Continuously divide the decimal number by ***sixteen*** while keeping track of the remainders, until reaching the result ZERO. (Remember: A=10, B=11, C=12, D=13, E=14, F=15).
- First remainder is the LSD (least-significant digit).
- Last remainder is the MSD (most-significant digit).

- Example:

$1234_{10} = ?_{16}$

16 | 1234		
16 | 77	2	
16 | 4	13 = D	
0	4	

$1234_{10} = 4D2_{16}$

- Exercises:

  What is the hexadecimal equivalent value of the following decimal numbers?

  1- 455_{10}

  2- 214_{10}

  3- 3509_{10}

4 FROM BINARY TO DECIMAL

- Each digit in a number has a position.
- Positions start by zero, and incremented by 1 from right to left.
- To convert from binary to decimal:
  - Multiply each binary digit by ***two*** to the power ***k***, where ***k*** is the position of the digit.
  - The final decimal value is the sum of the resulted multiplications.

- Example:

101011_2 =>

$1 \times 2^0 = 1$
$1 \times 2^1 = 2$
$0 \times 2^2 = 0$
$1 \times 2^3 = 8$
$0 \times 2^4 = 0$
$1 \times 2^5 = 32$

43_{10}

- Exercises:

  What is the decimal equivalent value of the following binary numbers?

  1- 11011001_2

  2- 101001_2

  3- 10101100_2

5 FROM OCTAL TO DECIMAL

- To convert from octal to decimal:
  - Multiply each octal digit by ***eight*** to the power ***k***, where ***k*** is the position of the digit.
  - The final decimal value is the sum of the resulted multiplications.

- Example:

724_8 =>

$4 \times 8^0 = 4$

$2 \times 8^1 = 16$

$7 \times 8^2 = 448$

468_{10}

- Exercises:

  What is the decimal equivalent value of the following octal numbers?

  1- 3427_8

  2- 125_8

  3- 71263_8

6 FROM HEXADECIMAL TO DECIMAL

- To convert from hexadecimal to decimal:
  - Multiply each hexadecimal digit by ***sixteen*** to the power ***k***, where ***k*** is the position of the digit. (Remember: A=10, B=11, C=12, D=13, E=14, F=15).
  - The final decimal value is the sum of the resulted multiplications.

- Example:
  Convert ABC_{16} to decimal.

$$ABC_{16} \Rightarrow C \times 16^0 = 12 \times 1 = 12$$
$$B \times 16^1 = 11 \times 16 = 176$$
$$A \times 16^2 = 10 \times 256 = 2560$$
$$2748_{10}$$

- Exercises:

  What is the decimal equivalent value of the following hexadecimal numbers?

  1- $D25E_{16}$

  2- $1FA_{16}$

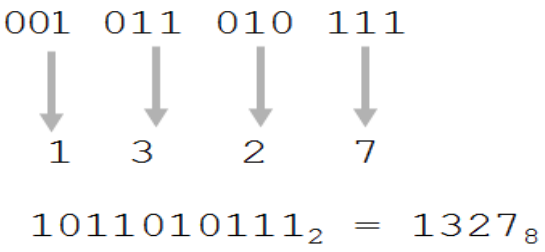
  3- $C4B8D_{16}$

7 FROM BINARY TO OCTAL

- Start from right to left, and split the binary number into groups of ***three*** bits.
- Add zeros to the left of the leftmost group if it has bits less than three.
- Give each group the decimal equivalent digit, as the table aside shows:

Binary	Decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

- Example:
  Convert 1011010111_2 to Octal.



$$1011010111_2 = 1327_8$$

- Exercises:

  What is the octal equivalent value of the following binary numbers?

  1- 11001100_2

  2- 10011101_2

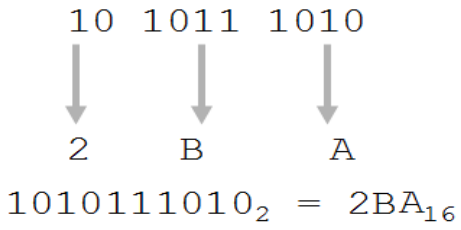
  3- 101010110111_2

8 FROM BINARY TO HEXADECIMAL

- Start from right to left, and split the binary number into groups of ***four*** bits.
- Add zeros to the left of the leftmost group if it has bits less than four.
- Give each group the decimal equivalent digit, as the table aside shows:

Binary	Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

- Example:
  Convert 1010111010_2 to hexadecimal.



- Exercises:

  What is the hexadecimal equivalent value of the following binary numbers?

  1- 011101100_2

  2- 1110010111_2

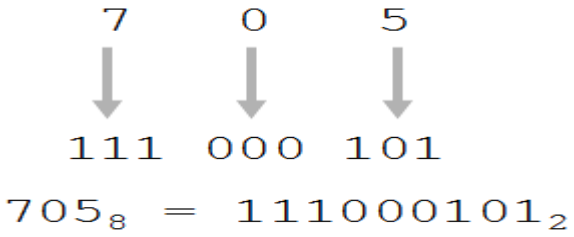
  3- 110101010110101_2

9 FROM OCTAL TO BINARY

- Substitute each octal digit by its ***three*** equivalent binary digits.

Binary	Decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

- Example:
  Convert 705_8 to binary.



- Exercises:

  What is the binary equivalent value of the following octal numbers?

  1- 2643_8

  2- 570_8

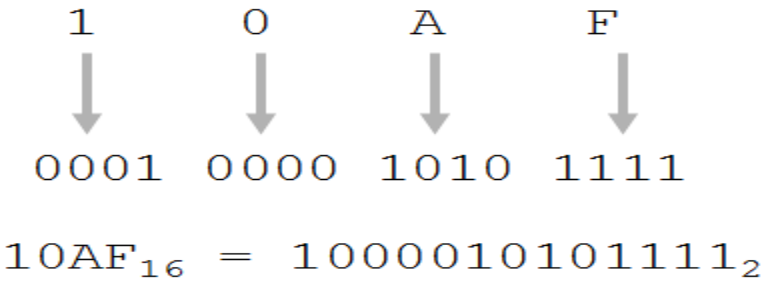
  3- 3416_8

10 FROM HEXADECIMAL TO BINARY

- Substitute each hexadecimal digit by its ***four*** equivalent binary digits.

Binary	Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

- Example:
  Convert $10AF_{16}$ to binary.



- Exercises:

  What is the binary equivalent value of the following hexadecimal numbers?

  1- $B68E_{16}$

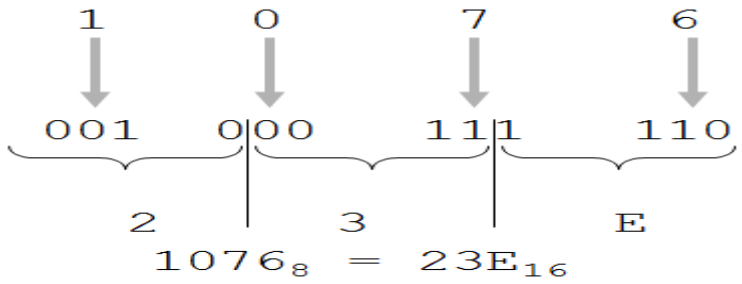
  2- 120_{16}

  3- $3D5C9_{16}$

11 FROM OCTAL TO HEXADECIMAL

- Use either binary or decimal as an intermediary numeral system:
  - Octal → Binary → Hexadecimal
  - Octal → Decimal → Hexadecimal

- Example:
  Convert 1076_8 to hexadecimal.



- Exercises:

  What is the hexadecimal equivalent value of the following octal numbers?

  1- 2057_8

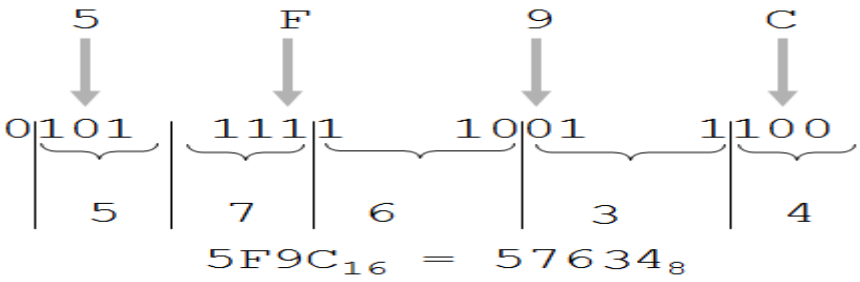
  2- 712_8

  3- 35674_8

12 FROM HEXADECIMAL TO OCTAL

- Use either binary or decimal as an intermediary numeral system:
  - Hexadecimal → Binary → Octal
  - Hexadecimal → Decimal → Octal

- Example:
  Convert $5F9C_{16}$ to octal.



- Exercises:

  What is the octal equivalent value of the following hexadecimal numbers?

  1- $B68E_{16}$

  2- 120_{16}

  3- $3D5C9_{16}$

BINARY ADDITION

- One of the basic (primitive) arithmetic operations the computer can perform is ***addition***.
- Adding two binary numbers is applied by adding their corresponding bits as follows:

0	0	1	1
0 +	1 +	0 +	1 +
0	1	1	1 0 (*with carry*)

- Examples: Perform the following arithmetic operations:

$1001_2 + 1101_2$

1001	9
1101 +	13 +
10110	22

$10111_2 + 100101_2$

010111	23
100101 +	37 +
111100	60

- **Exercises:**

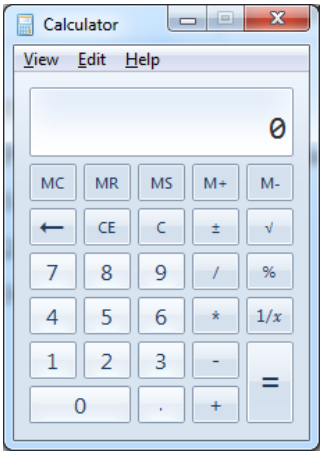
Perform the following arithmetic operation:

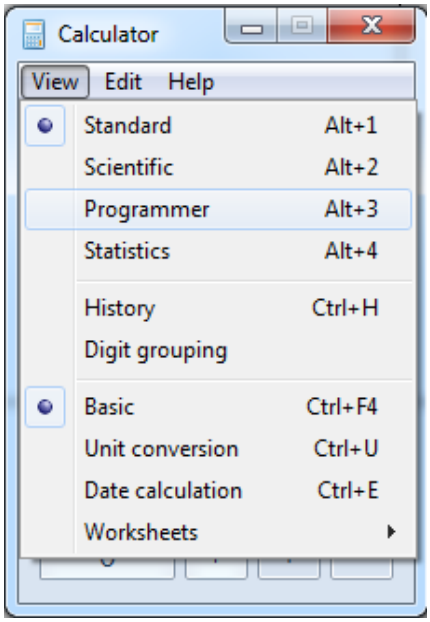
1- $10001_2 + 101010_2 = (\quad)_2$

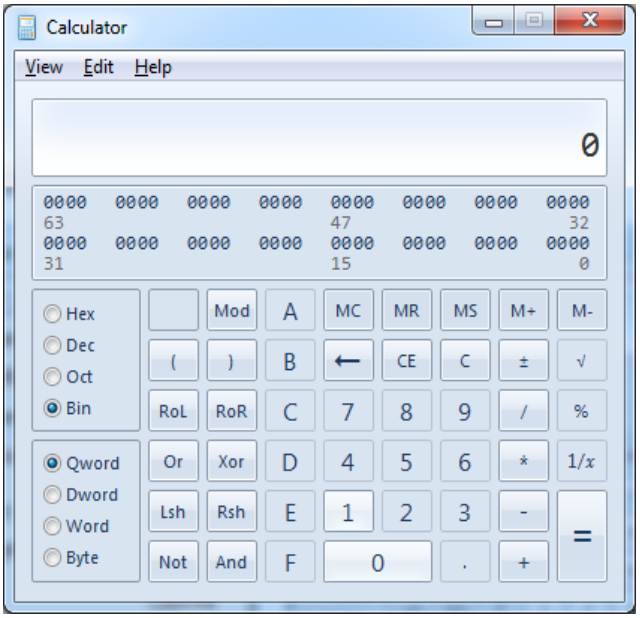
2- $45_{10} + 101010_2 = (\quad)_2$

3- $63_{10} + 27_{10} = (\quad)_2$

Activity: Using Windows® Calculator for Binary Addition.







HOW MANY DISTINCT CASES CAN BE FORMED FROM *n* BITS?
WHAT IS THE MAXIMUM VALUE CAN BE FORMED FROM *n* BITS?

Number of bits	Cases	How many cases?	Maximum value
	0	$2 = (2^1)$	$1 = (2^1 - 1)$
1	1		
	00	$4 = (2^2)$	$3 = (2^2 - 1)$
	01		
2	10		
	11		
	000	$8 = (2^3)$	$7 = (2^3 - 1)$
	001		
	010		
3	011		
	100		
	101		
	110		
	111		
	0000	$16 = (2^4)$	$15 = (2^4 - 1)$
4	0001		
	⋮		
	1111		
n		**2^n**	**$2^n - 1$**

جامعة بوليتكنك فلسطين

PROBLEM SOLVING TECHNIQUES

Prepared by:
Eng. Yousef Salah

This material developed under the objectives of FESTEM project funded by the EU.
https://festem.ps/

Co-funded by the Erasmus+ Programme of the European Union

CHAPTER SEVEN
PROBLEM SOLVING TECHNIQUES

PREFACE:

This chapter begins with a short introduction about programming languages. It presents the classification of languages by generations, and points out the nature of each category. This is followed by the software development life cycle while giving thorough description about each phase in the software creation process. The chapter also teaches students how to think algorithmically and solve problems efficiently. Next, several computational problems are solved based on the appropriate control structures and logic needed to achieve the solution.

INTENDED LEARNING OUTCOMES:

After completing this chapter students will be able to:

1) Identify the classifications of programming languages.
2) Discuss programming language translators.
3) Recognize the main phases of the software development life cycle.
4) Explain what a computer algorithm is, and how it can be represented.
5) Describe program logic and control structures used to formalize the solution of computational problems.

FURTHER READING:

1) Computer Science An Overview, 13th Eition, J Glenn Brookshear; Dennis Brylow, Pearson, @2019.
2) Introduction to the Design and Analysis of Algorithms, Levitin, A. V., 2nd ed. Boston, Addison-Wesley, 2007.
3) Computer Algorithms: Introduction to Design and Analysis, Baase S., 3rd ed. Boston, Addison-Wesley, 2000.

INTRODUCTION

- Computer hardware is nothing without being instructed to do tasks.
- Therefore, our goal is to give commands to the computer in order to perform specific tasks in certain order.
- **Computer Program**: a list of instructions (i.e. commands) that directs the computer to perform a specific task. These instructions are written in a special language called ***programming language***.
- **Programmer** (or developer): a person who writes and modifies computer programs.
- **Programming Language**: set of words, abbreviations, and symbols used to create computer programs.
- Any programming language has a grammar (set of rules) that governs writing valid statements in this language.
    - These rules are also called the **Syntax** of the language.

TYPES OF PROGRAMMING LANGUAGES

- The language of a computer is called the ***machine language***, where the commands are represented as sequences (i.e. patterns) of ZEROs and ONEs called bits.

  Example of a machine code: **1011010000000101**

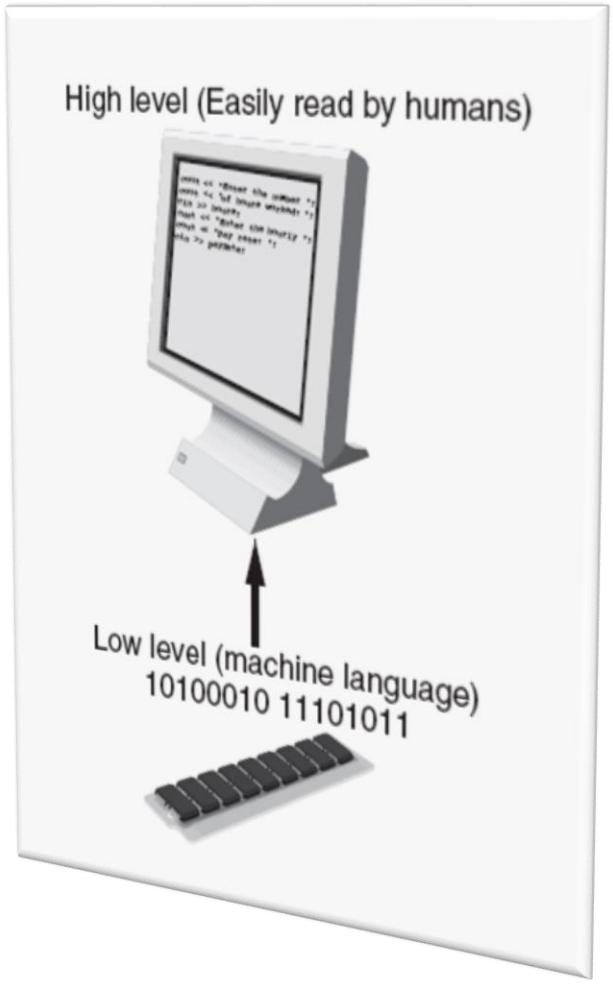
- However, not all machines (i.e. computers) have a global machine language.
- Each computer's designer chooses his own set of binary codes to perform the basic operations required from the computer.
- For example, IBM computer manufacturer may represent the Addition operation as **10101010**, whereas Apple computer manufacturer may choose **11110000** for the Addition operation.

- There are two types of programming languages: ***low-level*** and ***high-level*** programming languages.

- **Low-level Programming languages:**

  Their Syntax is close to the hardware. Low-level programming languages include:

  - Machine Language (First Generation)
  - Assembly Language (Second Generation)



- **High-level Programming languages:**

  Their syntax is more like human language, and use English-like words that are understandable by people. High-level Programming languages include:

  - Third Generation Languages (3GLs).
  - Forth Generation Languages (4GLs).
  - Visual Programming Languages.
  - Web Development Languages.

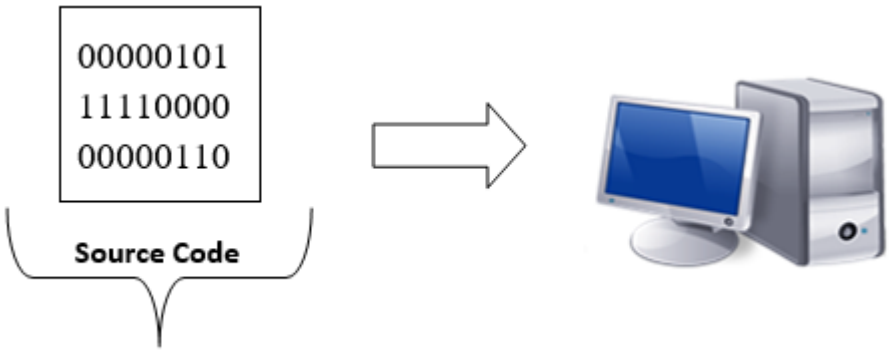
PROGRAMMING LANGUAGES CLASSIFICATIONS

1 Machine Language (First Generation)

- Programs written in a machine language are patterns (or sequences) of 0s and 1s.
- For simplicity, assume we have a computer machine that has the following instruction set:

Binary Pattern	Operation
11110000	Add two numbers
11001100	Subtract two numbers

The following machine program (***code***) instructs the computer to add the two numbers 5 and 6.



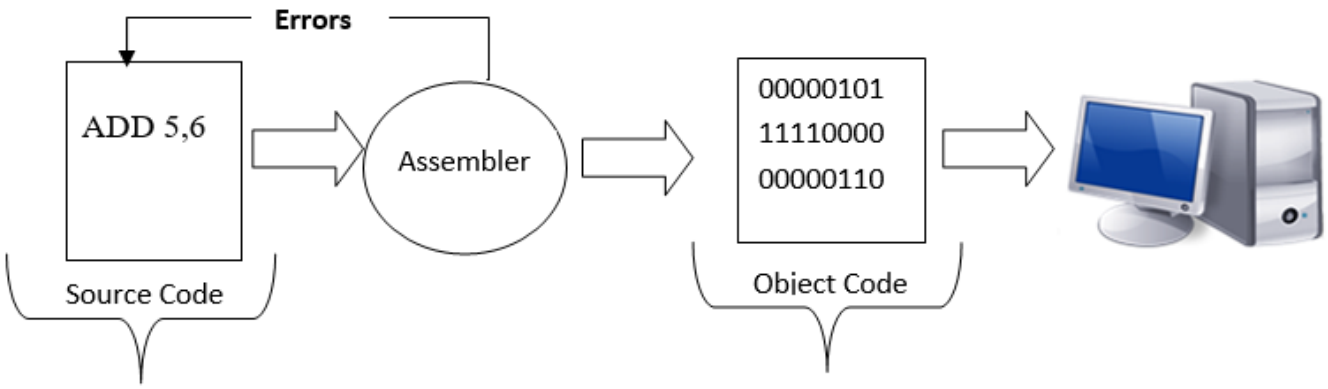
- Instructions (or statements) written in a programming language are called ***code*** or ***source code***.
- The commands are executed directly by the computer processor after loading into memory.
- The disadvantages of writing computer programs in machine languages are:
    - The programming is tedious, error prone and time-consuming.
    - Debugging (i.e. finding errors) is difficult.

2 Assembly Language (Second Generation)

- Instructions in a computer program written in an assembly language are meaningful and readable names and symbols (called mnemonics) that correspond to 0s and 1s.
- For example, for the computer machine discussed later, the following mnemonics are used:

Binary Pattern	Mnemonic	Meaning
11110000	ADD	Add two numbers
11001100	SUB	Subtract two numbers

The following assembly code instructs the computer to add the two numbers 5 and 6.



- Programs written in assembly language cannot be delivered directly to the computer.
- A system software called **Assembler** is a software that translates a program written in assembly language into its equivalent code in machine language called ***object code***.

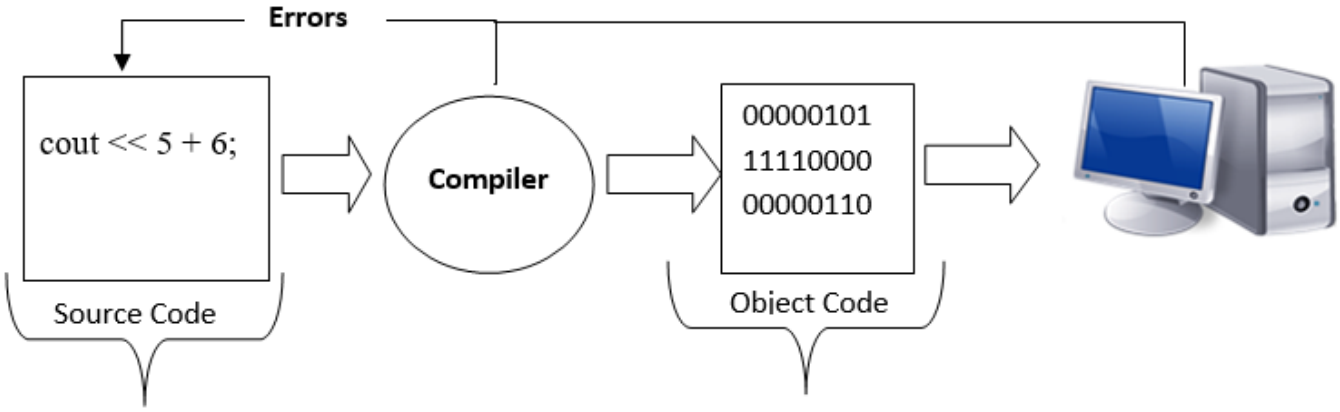
- The disadvantages of writing computer programs in assembly languages are:
    - Debugging is still difficult.
    - The translation process requires extra time.

3 Third Generation Languages (3GLs)

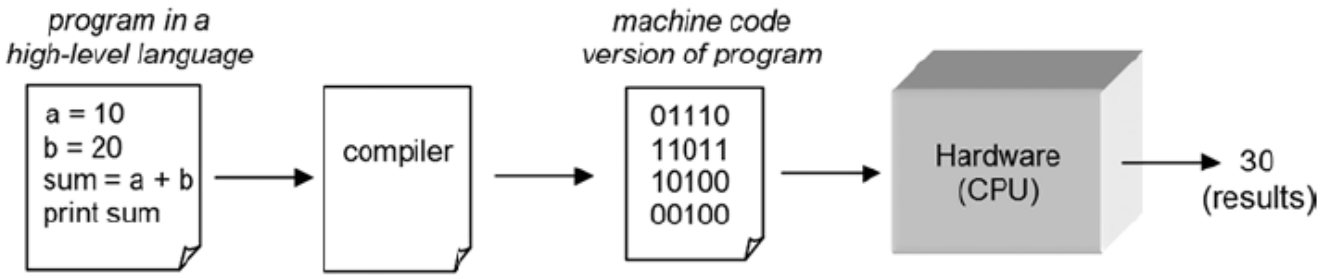
- Third Generation Programming languages are high-level languages since the commands and instructions used are human readable, and they use English-like keywords.
- For example, keywords like: ***include***, ***return***, ***while*** are used in C++ (a high level programming language).
- 3GLs also use symbols for arithmetic and logical operations like: **+, *, > , ?, !, …**
- Examples of Third-Generation programming languages: C, C++, Pascal, FORTRAN, COBOL, Java, C#, Python ...
- A computer program written in a high level programming language should be translated into machine language before being delivered to the computer.
- There are two types of translators:
    - Compiler
    - Interpreter

* COMPILERS:

- A Compiler: a program that translates instructions written in a high-level language into the equivalent machine language (i.e. Object Code).



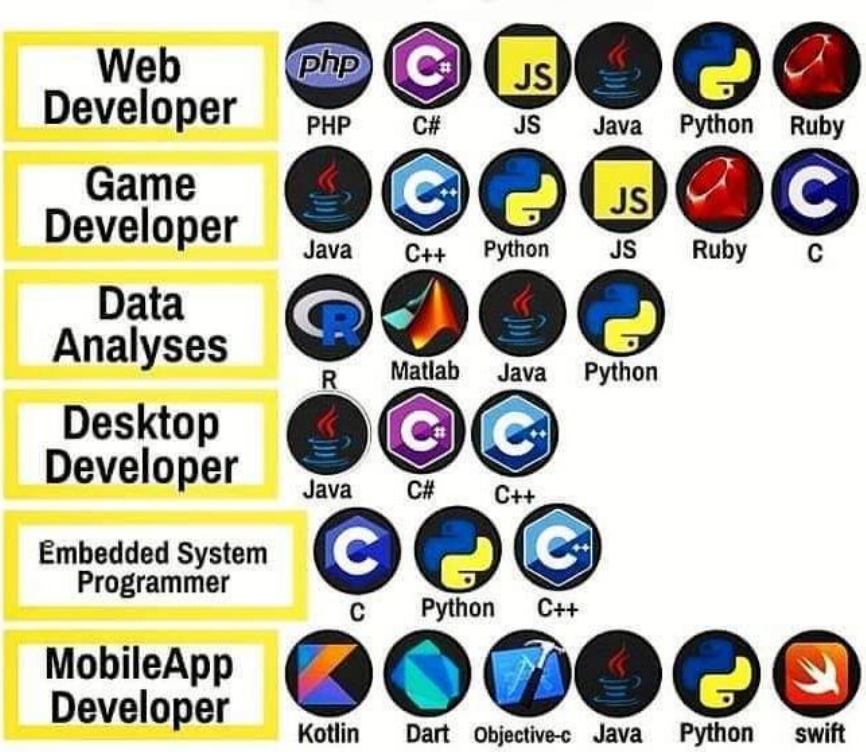
- A complier translates the entire source code into object code before the execution. So, the translation is done one time, and the object code is executed many times.



Web Development Languages

- There are special programming languages that are used to construct web pages.
- Examples of Web Page languages include:
    - HTML: Hypertext Markup Language
    - PHP: Personal Home Page Language
    - ASP: Active Server Pages Language
    - JSP: Java Server Pages Language

Programming Languages Based on Specialty:



Activity:

- What is the meaning of open source software?

- List open source OS examples for Desktop and Mobile devices.

Activity:

Visit https://www.w3schools.com/ website, and have a look to different common programming languages.

SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

- The creation of a computer program is a developmental process.
- A ***computer program*** solves a problem in a computerized way.
- A problem here means *computational problem*.

- ***Programming*** is a process of problem solving, and a programmer is a problem solver person who finds the solution of a problem using a problem solving technique.
- ***Computer Programming*** (also called ***software development***) is the process of planning, designing, creation, and testing a computer program.
- ***Computer Programmer*** (or Software Developer) is a person who creates computer software using a programming language.

- Creating a computer program resembles building a house. You need to perform some planning, designing, and implementation in order to build a house.
- The program development cycle is a series of steps the programmers follow in order to build a computer program.

- The software development life cycle (SDLC) consists of the following main phases:
    - Problem Analysis.
    - Program Design.
    - Program Coding (or Implementation).
    - Program Testing.
    - Program Operation and Maintenance.

PHASE 1: PROBLEM ANALYSIS

- Analyzing a problem (also called problem definition) involves studying the requirements needed from the problem. Sometimes this step is called ***Requirement Analysis***.

- The programmer (or system analyst) tries to understand the user's requirements through identifying the inputs, processing, and outputs of the problem.

Software Engineering:
Software Engineering is defined as a process of analyzing user requirements and then designing, building, and testing software application which will satisfy these requirements.

- The system analyst can then document these requirements in a neat and organized format. This ***documentation*** is the road map that will lead the next steps.
- The Inputs, Processing, and Outputs also called IPO are defined clearly in this phase. An IPO chart may appear in the documentation to model the inputs, outputs, and processing:
    - **Inputs:** the information, ideas, and resources.
    - **Processing:** the operations, events, and actions taken upon/using input.
    - **Outputs:** results of the processing.

Example:

Problem Statement: Determine the total cost of apples given the number of kilos of apples purchased, and the cost per kilo of apples.

- ***Problem Input:***
  - Quantity of apples purchased (in kilos)
  - Cost per kilo of apples (in dinars per kilo)
- ***Processing:***
  Total cost = Number of kilos of apples × Cost per kilo
- ***Problem Output:***
  - Total cost of apples (in dinars)

IPO Chart:

Input	Processing	Output
number of kilos of apples cost apples per kilo	Total cost = Number of kilos of apples × Cost per kilo	Total cost of apples

Exercise:

Problem Statement: Suppose that a store makes a discount on its items. What is the IPO chart that shows the amount of money that a customer has to pay when purchasing an item, where the following data are given: the price of the item before discount and the discount rate.

IPO Chart:

Input	Processing	Output

PHASE 2: PROGRAM DESIGN

- Designing a problem involves describing the solution of the problem. The solution lists the ***main steps*** required to be carried out in order to reach output(s) from input(s).
- The designing process involves finding an ***algorithm*** that satisfies the requirements.
- An ***algorithm*** is a step-by-step procedure that describes the solution of a problem to perform a specific task.

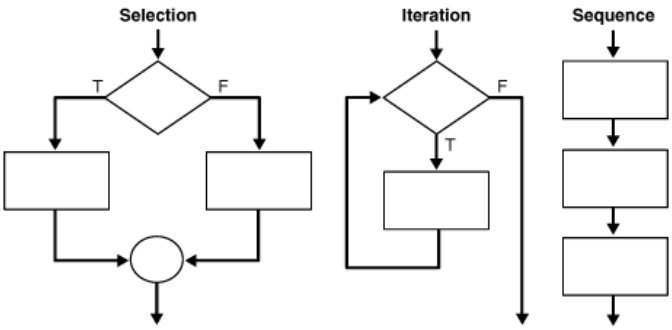
Algorithm (الخوارزمية):

The word "Algorithm" derived from the name of the Arab Muslim mathematician scientist named: "Muhammad Al-Khwarizmi.", who is the founder of "Al-jabr", in which he introduced the fundamental algebraic methods and techniques for solving equations.

- (Other Definition) An algorithm is a sequence of a finite number of steps arranged in a specific logical order which, when executed, produces the solution for a problem.

- An algorithm is an abstract and general description. This means that writing an algorithm means listing the main steps that solve the problem **without** identifying the **details** of implementation (i.e. programming language, compiler, operating system, …).

- The solution algorithm is also called the ***program logic***, since it shows the logical order of execution.
- The order in which statements (i.e. instructions) are executed is called program flow control (or flow of control).
- There are three ways (logic structures or control structures) that illustrate this order:
    - Sequence control structure
    - Selection control structure
    - Repetition control structure



- There are many ways used to represent algorithms, some are textual, others are graphical.
- We'll study two design tools (methodologies) that are used to express an algorithm:
    - Pseudocode
    - Flowcharts

* PSEUDOCODE:

- English-like statements that describe the steps that solve the problem.
- Pseudocode uses keywords similar to those existed in high level languages.

Example:

Write an algorithm to determine a student's final grade and indicate whether it is passing or failing. The final grade is calculated as the average of four marks.

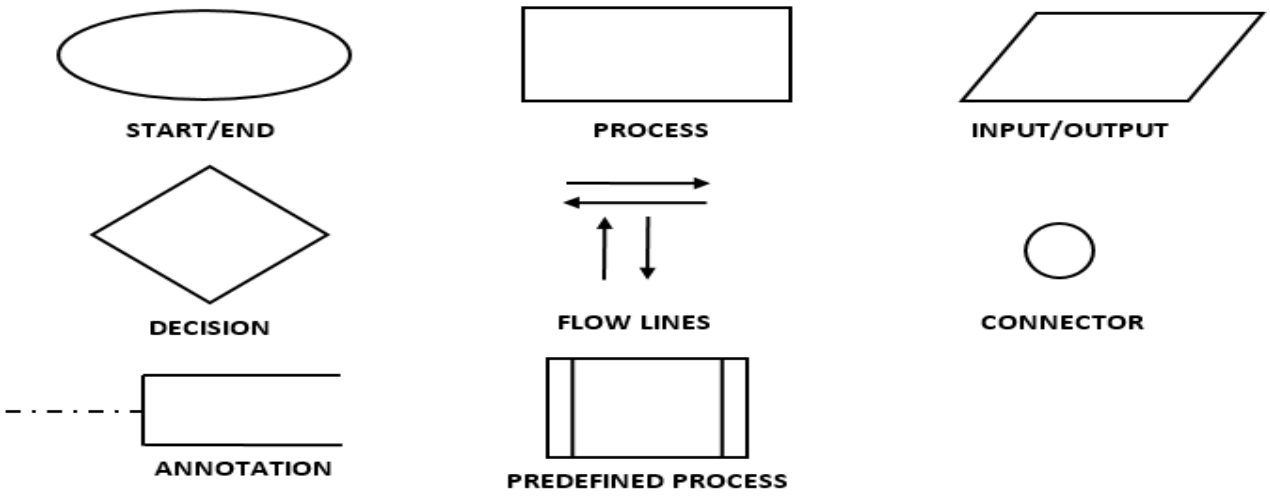
Solution described in Pseudocode:

- *Input a set of 4 marks*
- *Calculate their average by summing and dividing by 4*
- *if average is below 50*
  Print "FAIL"
  else
  Print "PASS"

- No strict rules govern writing the Pseudocode. For instance, we can write the word **"otherwise"** instead the word **"else"** in the code above.
- You can write the algorithm in more detailed fashion like:
  - Step 1: READ: M1,M2,M3,M4
  - Step 2: grade ← (M1+M2+M3+M4)/4
  - Step 3: IF (grade < 50) THEN
    PRINT "FAIL"
    ELSE
    PRINT "PASS"

* FLOWCHART:

- A flowchart is a ***graphical representation*** of the sequence of operations in an algorithm.
- A flowchart shows the ***logic*** of an algorithm. This logic is the order in which operations are performed to reach the goal.
- The flowchart of an algorithm graphically shows the three logical order of the execution: Sequence, Selection, and Repetition.
- The symbols (i.e. constructs) of a flowchart are graphical symbols. These constructs collectively are called the ***notation*** of the flowchart methodology. Here is the description of these symbols:



PHASE 3: PROGRAM CODING (OR IMPLEMENTATION)

- ***Program coding*** (or ***development***) is the process of converting the designed solution algorithm into program instructions written in a particular programming language.
- Mostly, the programmer chooses a programming language to write the instructions, and saves these instructions into a file. This version of a file is what we called the ***source code***.
- Actually, the programmer needs a text editor to type the instructions. Then a language translator is used to perform the source code translation into machine language version called the ***object code***.
- We had already discussed the three common types of language translators: Assemblers, Compilers, and Interpreters.

- Nowadays, most of the programming languages use a single interface for all the programs and development tools required in the development process (like text editor, translator, debugger and other utilities). These programs are referred to as an **Integrated Development Environment (IDE)**, or **Software Development Kit (SDK)**.

Well-known IDEs are:
- Visual Studio (from Microsoft®).
- Android Studio.
- Visual Studio Code.
- Eclipse.

PHASE 4: PROGRAM TESTING

- After creating the source code, the programmer tests the following:
    - Make sure that the ***application*** works correctly, and it is free of errors (or bugs).
    - Make sure that all the user requirements are satisfied.
- An error can be one of the following two types:
    - **Syntax Error:** occurs when the code violates the syntax of the programming language. The translator issues an error message so that the programmer can correct it.
        - For example: entering "prnt" instead of "print".
    - **Logic Error (or Semantic Error):** occurs during the run-time of the program due to a mistake in the solution algorithm, and may cause the program to terminate, hang, or behave wrongly.
        - For Example: finding the average of three numbers n1, n2 and n3 using the equation: (n1+n2+n3)/2.0.
- **Debugging:** The process of locating and correcting syntax and logic errors in a program.

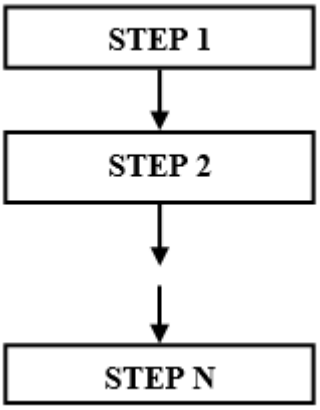
PHASE 5: PROGRAM OPERATION AND MAINTENANCE

- After testing the application software, it is released to users for operation.
- ***Program Operation*** involves installing the software, and allowing the users to use it.
- Program Maintenance involves:
    - Fixing any emerging errors or faults that did not appear during the testing process.
    - Adding new services and functionalities to the released software.

PROGRAM LOGICAL ORDER (CONTROL STRUCTURES)

* SEQUENCE CONTROL STRUCTURE:

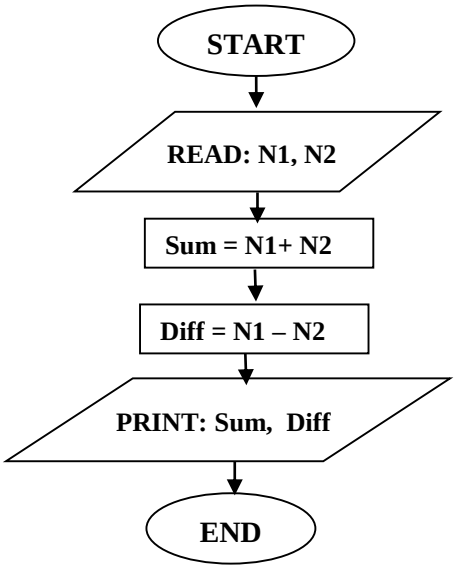
- The steps of the algorithm are carried out one after another.



Example:

Use Flowchart and Pseudocode techniques to describe the algorithm that reads two numbers, calculates and prints their sum and difference.

FLOWCHART



PSEUDOCODE

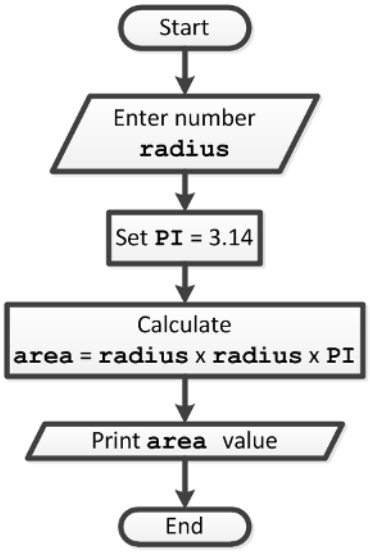
BEGIN
- INPUT: N1,N2
- Sum ← N1+ N2
- Diff ← N1- N2
- PRINT: Sum, Diff

END

Example:

The following Flowchart and Pseudocode describe the algorithm that reads a radius of a circle, then finds its area ($Area = \pi \times radius^2$).

FLOWCHART



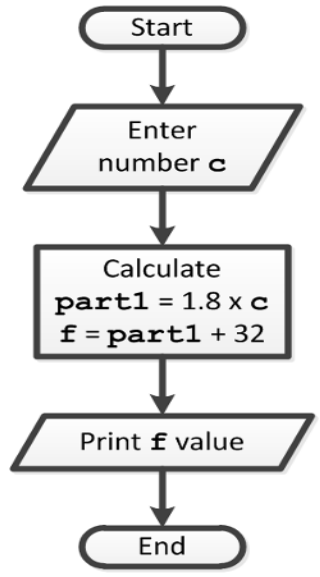
PSEUDOCODE

```
START
   Enter number radius
   Set PI = 3.14
   Calculate area = radius × radius × PI
   Print area value
END
```

Example:

The following Flowchart and Pseudocode describe the algorithm that converts the Celsius temperature to Fahrenheit temperature, using the equation: F = 1.8C + 32

FLOWCHART



PSEUDOCODE

```
START
        Enter number c
        Calculate part1 = 1.8 × c
        Calculate f = part1 + 32
        Print f value
END
```

Activity:

Use both flowchart and Pseudocode strategies to describe the algorithm that accepts three marks for a student, and then displays their sum and average.

Activity:

Suppose that a store makes discount on its items. Design an algorithm (using Flowchart and Pseudocode) that calculates the amount of money that a customer has to pay when purchasing an item, adding to it the required tax, where the following data are given: the price of the item before discount, the discount rate, and the tax rate. (Assume the discount is calculated after considering the TAX.)

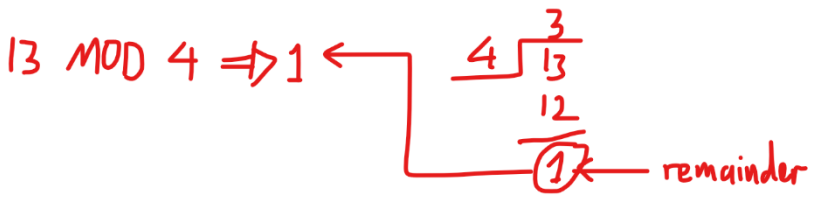
The Modulus Operation:

- When dividing a number by another one, a remainder is resulted.
- The Modulus operation (we'll refer to it as MOD) is used to obtain the resulted remainder after division.
- This operation usually helps when we are handling sub-parts of a particular number. For instance, we may determine whether a number is ODD or EVEN, or we may detect if it is a PRIME number or not, and the like.

Example:

13 MOD 4 = 1

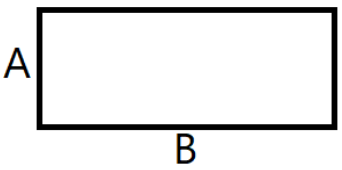
37 MOD 5 = 2



Exercises:

Use flowchart and Pseudocode strategies to design an algorithm that solves each of the following problems:

4) Finding the area and perimeter of a rectangle, assuming the length and width (A and B) are given as an inputs.

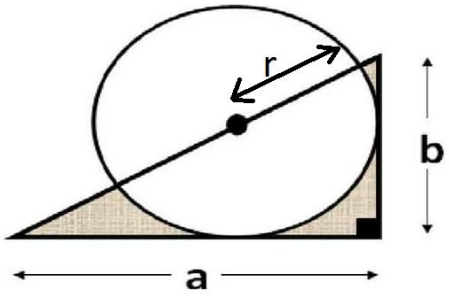


5) Calculating the percentage of increment in the salary of an employee, when inputting the salary before increment, and the value of increment.

6) Reading two numbers N1 and N2 and swapping their values.

7) Reading a number that represents an elapsed time in seconds. The algorithm displays how many hours, minutes and seconds this number contains. **For example,** if the number of input seconds is 8500, then the output is: 2 Hours, 21 Minutes, 40 Seconds.

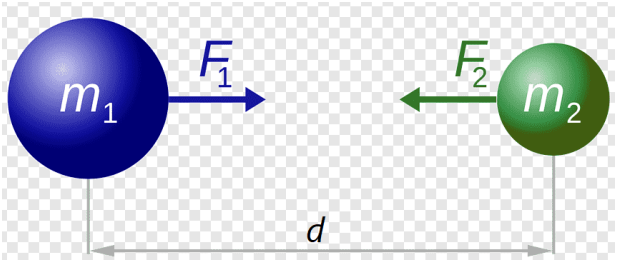
8) Given a, b and r as inputs, write the algorithm that will calculate the total area of the shaded parts in the figure.



9) Reading a *three-digit* number and printing each digit separately from right to left.
If the number is 479, the algorithm will print: **9 7 4**

10) Newton's law states that the force F, between two bodies of masses M1 and M2 is given by:

$$F = k\left(\frac{M_1 M_2}{d^2}\right)$$



in which k is the gravitational constant and d is the distance between the bodies. The value of k is 6.67 X 10^{-8} dyn.cm²/g². Design an algorithm that prompts the user to input the masses of the bodies, and the distance between the bodies. The algorithm calculates and outputs the force between the bodies.

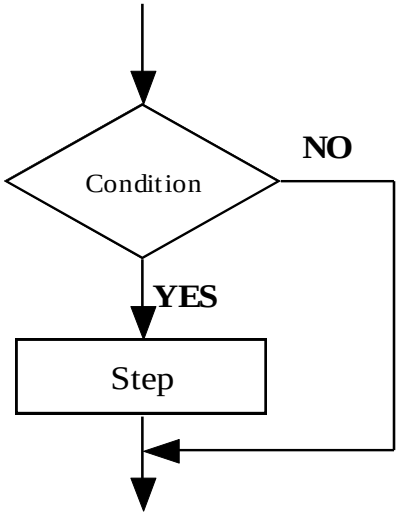
* SELECTION CONTROL STRUCTURE:

- A ***Selection*** (or ***Branching***) control structure shows which steps should be performed, and which should be ignored based on the evaluation of certain *condition*.
- A *condition* is a question that has an answer either **YES** (i.e. TRUE) or **NO** (i.e. FALSE).
- The condition (or ***logical condition***) is created by using logical or arithmetic operations like: $+, -, \times, \div, >, <, =, \neq, \leq, \geq$ …
- Examples of conditions:
  - *month* = 2 ?
  - *average* ≥ 90.0 ?
  - (1000 > *salary*) and (*rank* = 'A') ?

- There are three ways of using Selection Control Structure:
  - *if-then control structure*
  - *if-then-else control structure*
  - *case selection control structure*

if-then control structure:

- If the value of the condition is TRUE the following step(s) will be executed, otherwise (if it's FALSE) the step(s) will be ignored.



The *if-then* Control Structure

- Pseudocode of the if-then structure:

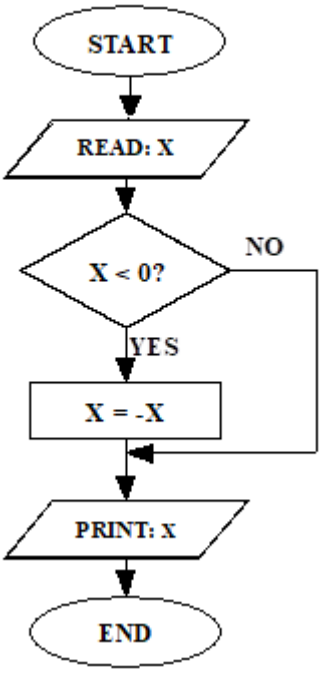
```
If( Condition ) Then
        Statement
```

Example:

Design an algorithm that reads a number and displays its absolute value.

$$|X| = \begin{cases} +X & X \geq 0 \\ -X & X < 0 \end{cases}$$

FLOWCHART



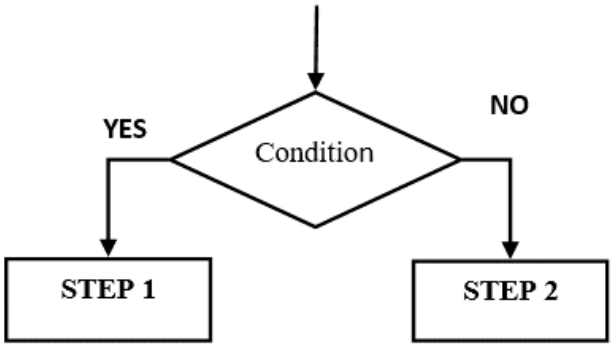
PSEUDOCODE

```
START
        READ:  X
        IF X < 0 THEN
            X = -X
        PRINT: X
END
```

if-then-else control structure:

- If the condition results **YES** the algorithm performs a specific action, otherwise if it results **NO** the algorithm performs a different action.
- Pseudocode of the if-then-else structure:

```
IF( Condition ) THEN
        Statement1
ELSE
        Statement2
```

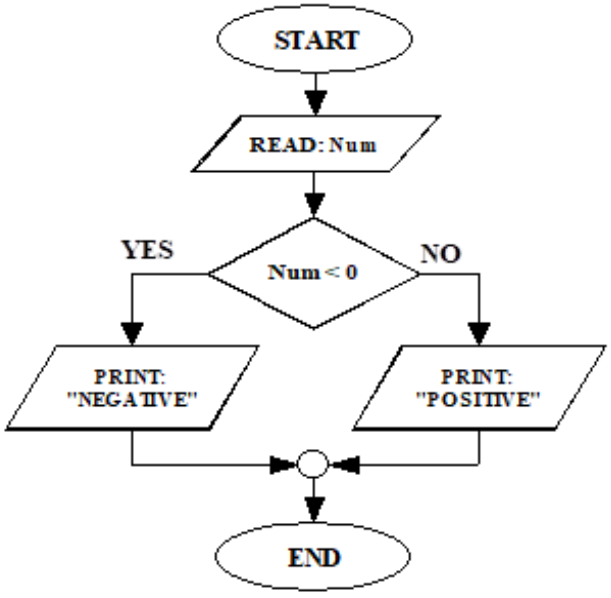


The *if-then-else* Control Structure

Example:

Design an algorithm that reads a number and displays whether it's POSITIVE or NEGATIVE.

FLOWCHART



PSEUDOCODE

BEGIN
- READ: X
- IF (X < 0) THEN
  PRINT "NEGATIVE"
  ELSE
  PRINT "POSITIVE"

END

Example:

Design an algorithm that reads a number and displays whether it is ODD or EVEN.

Example:

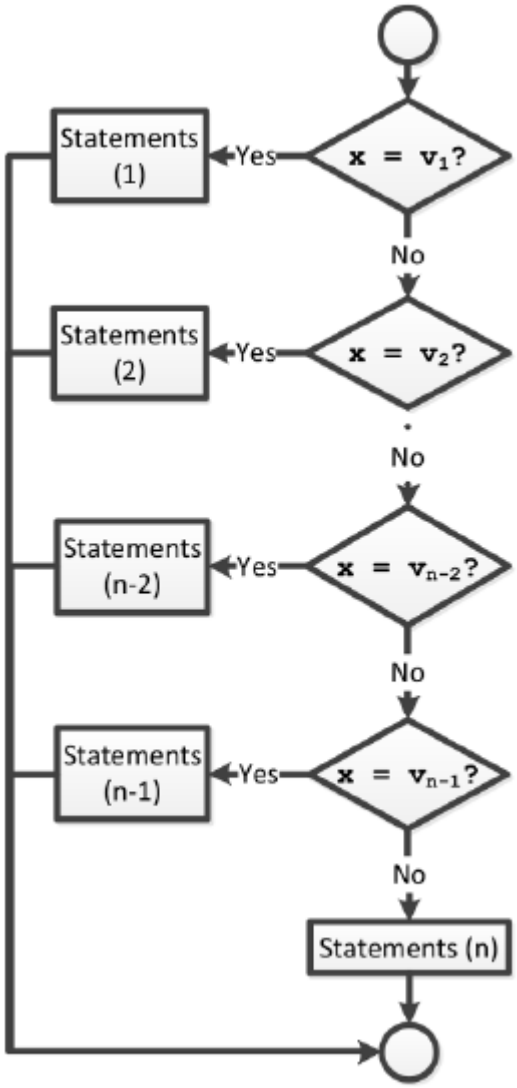
Design an algorithm that reads three numbers, then find and output the maximum entered value.

The case selection control structure:

- Multiple conditions are used to test where a single value matches some case.

- Pseudocode for the case structure:

```
IF( Condition1 ) THEN
        Statement1
ELSE IF( Condition2 ) THEN
        Statement2
ELSE IF( Condition3 ) THEN
        Statement3
        ⋮
```

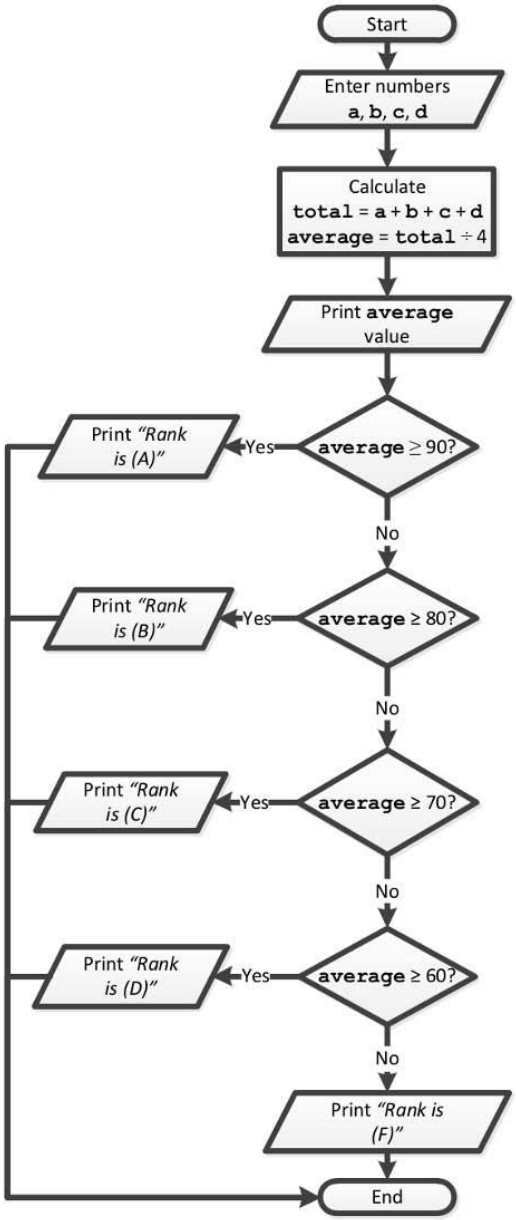


Example:

Write an algorithm, that reads four grades of a student, and output the average, then output the student rank, based on the following table:

Average Domain	Rank
90 and more	A
80 and more	B
70 and more	C
60 and more	D
Less than 60	F

FLOWCHART



PSEUDOCODE

Start
Enter numbers a, b, c, d
Calculate $total = a + b + c + d$
Calculate $average = total \div 4$
Print $average$ value
If ($average \geq 90$) Then
    Print "Rank is (A)"
Else If ($average \geq 80$) Then
    Print "Rank is (B)"
Else If ($average \geq 70$) Then
    Print "Rank is (C)"
Else If ($average \geq 60$) Then
    Print "Rank is (D)"
Else
    Print "Rank is (F)"
End

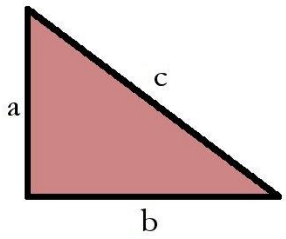
Example:

Write an algorithm, that reads a character, then will determine whether it is VOWEL letter or not. Note: Vowel letters are: A, E, I, O, U.

Exercises:

Use Flowchart and Pseudocode strategies, design an algorithm that solves each of the following problems:

1) The inputs are three numbers (a, b and c), representing the measurements of a triangle, the output should indicate if the inputs can form a right triangle (has the largest angle = 90), this can be done by using well-known equation: ($\mathbf{c^2 = a^2 + b^2}$).



2) In some universities, the student is full-time, when he registers (12) credit hours or more, or part-time if he registers less than (12) credit hours. Use the credits as input to print the tuition fees for that student, given that full-time fees are fixed at (2000), and the part-time fees are calculated as (the number of credits × 175).

3) The input is the total money spent at the mall, and the output is the number of points earned. For all customers, the awarded points are (1 point per 1 spent dollar), and if a customer spends more than (200) dollars, then he will get (30) more points.

4) The input is the student average in the school's final year, the output should be (Pass) if the grade is greater than or equal to (50%), or (Fail) otherwise. Also, as an additional output, the word (Awarded) should be printed if the average was (95%) or more.

5) Input the birthdate of a person (***bd/bm/by***) and the current date (***cd/cm/cy***), and calculate and display his age. (Assume the month to be 30 days).

6) Prompting the user to enter three integer numbers, and then displaying the numbers in ascending order.

7) Assume four marks were entered, if all the marks in the same range (e.g. between 0 and 100), write the algorithm that will calculate the average of the top three marks.

8) Read the coefficients *a*, *b* and *c* of the ***quadratic equation:*** ax^2+bx+c, then calculate and display the roots of the equation.
   - When the Discriminant $b^2 - 4ac$ is negative, then NO real roots.
   - When the Discriminant $b^2 - 4ac$ is ZERO, then there is single root ($r = -b / 2a$).
   - When the Discriminant $b^2 - 4ac$ is positive, then there are two roots $r1$ and $r2$:

$$r1, r2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

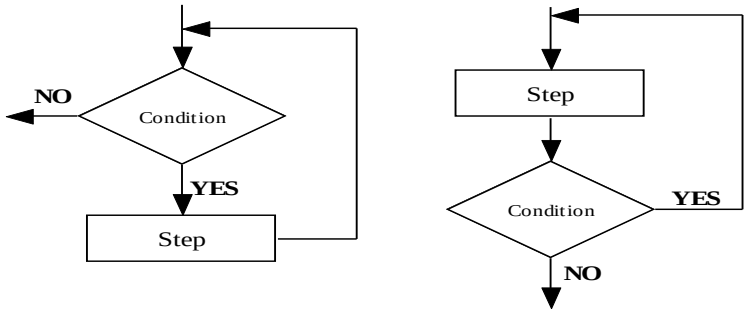
9) **Calculator Algorithm:** Input two numbers and the operation to be performed. The output is the result of applying the operation on the two numbers. The algorithms supports Addition ('+'), Subtraction ('-'), Multiplication ('*'), and Division ('/').

10) The input is one number representing base salary (BS), the output should be the amount of net salary (NS), which is calculated using the equation: (NS = BS + Bonus). The bonus is calculated by multiplying the base salary with its respective bonus rate. Bonus rates are listed in the following table:

Base Salary Domain	Bonus Rate
Less than (1000)	0.015
1000 - 1499	0.020
1500 - 1799	0.030
1800 - 2099	0.055
(2100) and more	0.075

* REPETITION CONTROL STRUCTURE:

- A ***Repetition*** (or ***Looping***, or ***Iteration***) control structure shows one or more steps that are being carried out repeatedly as long as some condition is being satisfied.



Possible *Looping* Control Structures

- Types of Loops:
  - Counter-Controlled: a counter controls the end of loop.
  - Event-Controlled: mostly, a special value stops the loop.

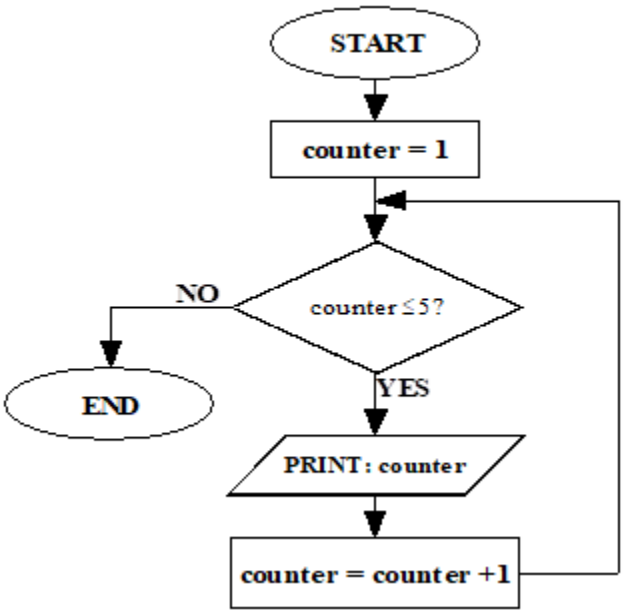
* COUNTER-CONTROLLED LOOPS:

- Iterates fixed number of times.

Example:

Write an algorithm that prints the numbers from 1 to 5 in an ascending order.

Tracing the flowchart



Example:

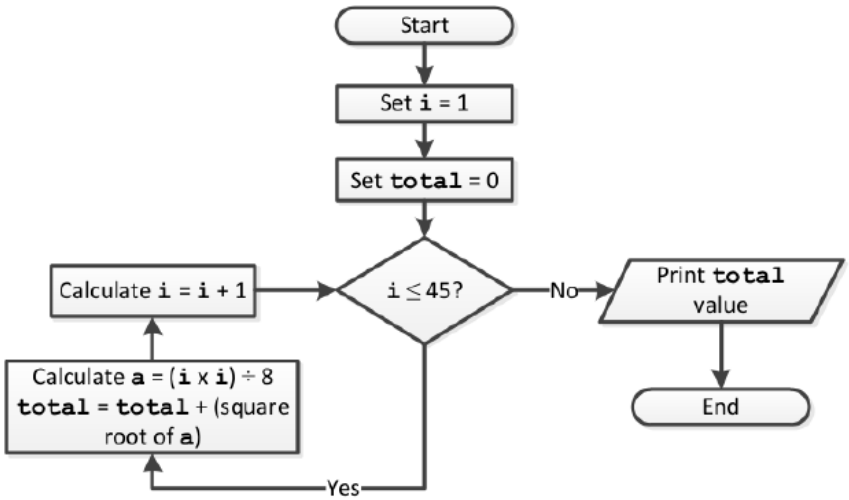
Write an algorithm that reads four numbers, calculates their sum and average, and prints output the results. (***Trace the algorithm by applying random input numbers***).

Example:

Write an algorithm that reads four numbers, calculates and prints the sum of EVEN entered values.

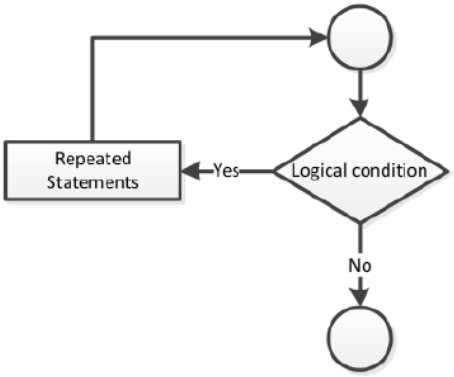
Example:

Trace the following algorithm and find the output.
What is the mathematical formula this algorithm represents?



- The **WHILE** keyword is used in the Pseudocode to indicate Looping.

WHILE(*Condition*)
Statement

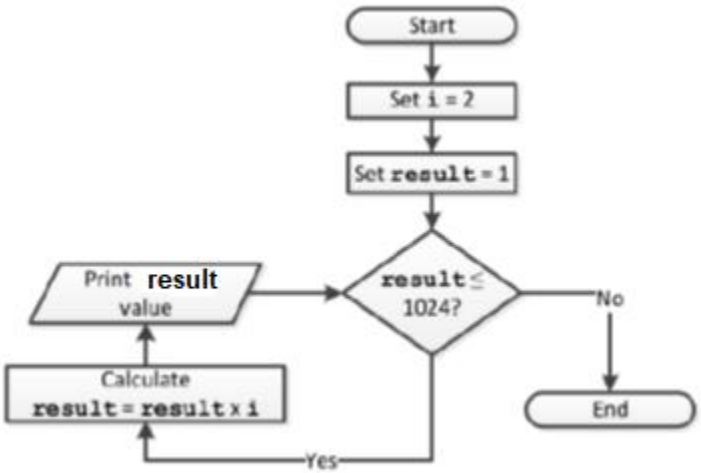


Example:

Design an algorithm that will print the following sequence:

2 4 8 16 32 64 128 256 512 1024 2048

Flowchart:



Pseudocode:

Start
Set $i = 2$
Set $result = 1$
While $(result \leq 1024)$
    Calculate $result = result \times i$
    Print $result$ value
End

* EVENT-CONTROLLED LOOPS:

- Iterates **UNTIL** a special value is encountered, or when a special action happens.

Example:

Design an algorithm that will continuously read characters until **'#'** is entered. The output is how many uppercase letters were entered.

Example:

A customer has 500 NIS cash money and wanted to buy some items from a market. Design an algorithm that will continuously read the prices of items until their sum exceeds the 500 NIS, then output how many items can be bought with this amount of money.

Exercises:

Use Flowchart and Pseudocode strategies to design an algorithm that solves each of the following problems:

1) Read eight numbers and print how many negative numbers were entered.

2) Read a number and calculate and display the sum of its divisors.
   Example: if the user inputs (15) then the divisors will be: (1, 3, 5, 15) and the output will be their total (24).

3) Input five numbers and determine the maximum one.

4) Read two integers **X** and **n**, and prints the value of **X** raised to **n** (i.e. $\mathbf{X}^{n}$).

Note: $\mathbf{X}^{n} = \underbrace{\mathbf{X * X * X * \dots X}}_{n \text{ times}}$

5) Input a positive number **N** and prints out its factorial (i.e. **N!**)

Note: $\mathbf{N!} = \underbrace{\mathbf{N * (N-1) * (N-2) * \dots * 2 * 1}}_{n \text{ times}}$

6) Compute the total of the digits, for some entered integer. An integer is a number without fraction.

7) Display a positive integer number, which is entered by the user, in reversed order.
   Example: when the input is (2753), the output will be (3572).
   Hint: use remainder and division by (10).

8) Compute the sum of all integers between two entered values (n1) and (n2), given that (n1) should be greater than (n2).

9) Find the greatest common divisor (GCD) of two positive integers. If any of these numbers is not positive, print a suitable message.

10) The user enters as much numbers as possible, this should continue if the input was not negative. After a negative number is entered, the input stops, and the output should be the count of the entered values. Do not include the last negative value into the count.
Example: if the entered values were (1, 6, 0, 4, 8, 11, 36, 2, -10), the output should be (8).

11) The user enters a positive integer (n), this is followed by entering more (n) numbers, and then output their average. Example: if the user first enters (5) to be stored as (n), then he should enter (5) other values, like: (9, 7, 3, 6, 10), the total of these values = 9 + 7 + 3 + 6 + 10 = 35, and their average is (35 ÷ 5) = 7, so the output is (7).

12) Write an algorithm that prompts the user to enter an integer number, and prints the digits of this number on separate lines: For Example: if the entered number is 65082 your program displays:

2
8
0
5
6

13) Enter an integer number and determine whether it's a PRIME number or not. (*Note:* a PRIME number only divides on itself and 1).

CHAPTER QUESTIONS:

Q1) State whether each of the following statements is **TRUE** or **FALSE**.

	Statement	TRUE / FALSE
[1]	Semantic errors are discovered during program run-time.	
[2]	Pascal is a low-level programming language.	
[3]	Instructions in assembly language are meaningful words called mnemonics.	
[4]	Pseudocode will not follow strict lexical rules when forming an algorithm.	
[5]	A language translator transforms machine code into source code.	
[6]	Program logic may combine different control structures.	
[7]	It is easy for programmers to write computer commands using machine languages.	
[8]	An interpreter will not generate an object code file for later execution.	
[9]	HTML is mainly used for game development.	
[10]	An algorithm should specify all the details of implementation.	
[11]	Debugging is the process of identifying user requirements.	
[12]	An algorithm is a general description of main steps that solve a problem and include details of the implementation	
[13]	Finding the algorithm that satisfies the program requirements is done as part of the design step.	
[14]	The Assembler translates a high-level program to its equivalent machine language code.	

Q2) Choose the best answer:

1	In which phase of SDLC does the software developer analyses whether software can be prepared to fulfill all the requirements of the end user?	**A.** Design **B.** Development **C.** Testing **D.** Planning
2	A program must be converted to _____ language to be executed by a computer.	**A.** Assembly **B.** Machine **C.** High level **D.** Very high level
3	A _____ error does not prevent the program from running, but causes it to produce incorrect results.	**A.** syntax **B.** hardware **C.** logic **D.** fatal
4	A(n) _____ is a set of well-defined logical steps that must be taken to perform a task.	**A.** logarithm **B.** plan of action **C.** logic schedule **D.** algorithm
5	An informal language that has no syntax rules and is not meant to be compiled or executed is called _____.	**A.** faux code **B.** pseudocode **C.** Python **D.** a flowchart
6	A _____ structure can execute a set of statements only under certain circumstances.	**A.** sequence **B.** circumstantial **C.** decision **D.** boolean
7	A _____ -controlled loop repeats a specific number of times.	**A.** event **B.** condition **C.** decision **D.** count

جامعة بوليتكنك فلسطين

COMPUTER PROGRAMMING USING C++

Prepared by:
Eng. Yousef Salah
Dr. Mohammad Abu Taha

This material developed under the objectives of FESTEM project funded by the EU.
https://festem.ps/

Co-funded by the Erasmus+ Programme of the European Union

CHAPTER EIGHT
COMPUTER PROGRAMMING USING C++

TEXTBOOK: C++ Programming: From Problem Analysis to Program Design, 6th Edition
AUTHOR: D.S. Malik

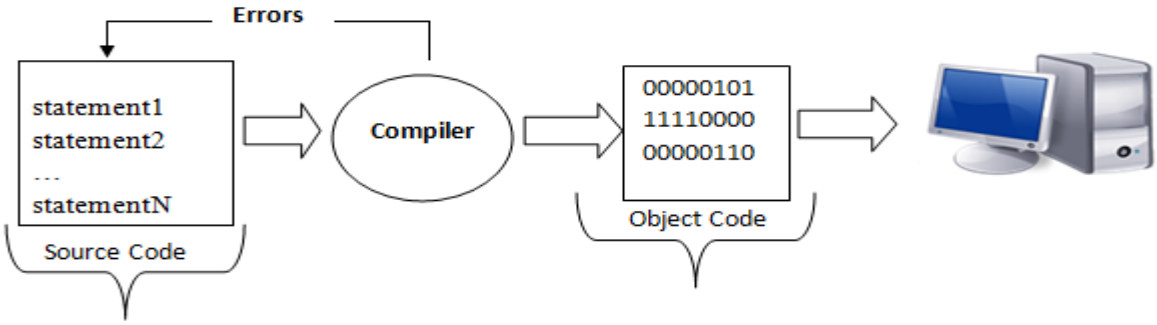
INTRODUCTION

- A Programming Language: set of words, symbols and abbreviations used to construct a computer program.
- A computer program is a collection of commands (a.k.a. instructions, or statements) that directs the computer to do a task.

- A Programming Language has:
  - Syntax: grammar rules used to construct valid (i.e. correct or legal) statements.
  - Semantics: meaning of statements.

C++ Programming Language

- C++ is a high level general-purpose programming language, which was developed by Bjarne Stroustrup at Bell Labs since 1979, as an extension of the C language.
- C++ is a compiled language, which means that it has a compiler.
- Many compilers were available by early 1990s, this let American National Standard Institution (ANSI) and International Standard Organization (ISO) to standardize the syntax of C++.
- We will study ANSI/ISO Standard C++ as it was standardized in mid-1998.
- A compiler translates source code into its corresponding machine code (or object code).



- Examples of C++ compilers, or sometimes called IDE (Integrated Development Environment): Dev C++, MS Visual C++, Eclipse, NetBeans, code::Blocks, Turbo C++, Borland C++, GCC.
- Web-based compilers for different programming languages:
  https://www.onlinegdb.com/online_c++_compiler

Rules and Syntax of C++

❖ C++ is a case-sensitive language.
This means C++ differentiates between lowercase and uppercase letters.
A ≠ a
Total ≠ total

By the way:
{ } are called Braces
( ) are called Parenthesis
< > are called Triangle Brackets
[ ] are called Brackets

❖ The basic command in C++ is called a "Statement".
Each statement in C++ ends by a semicolon **;**

❖ A C++ block is a set of statements enclosed between braces { }

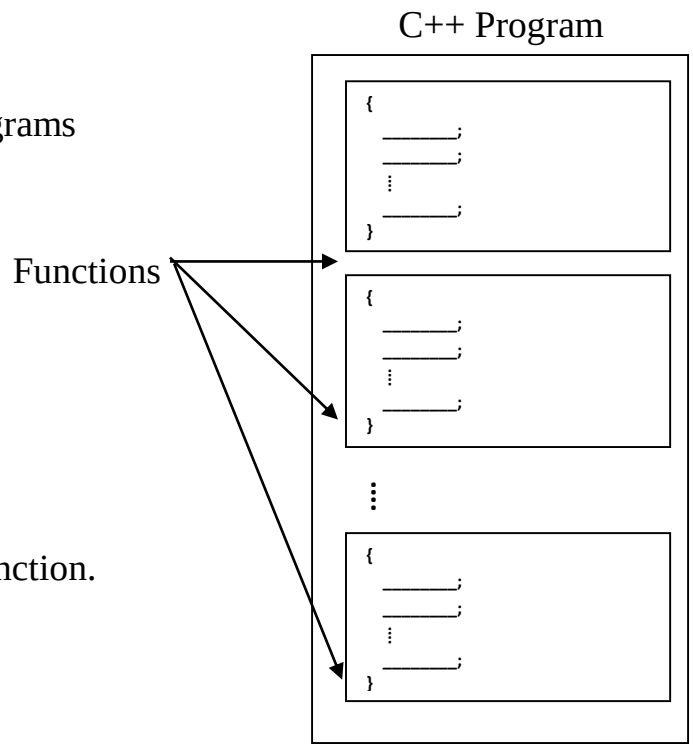
```
{
   _______;
   _______;
   ⋮
   _______;
}
```

❖ A C++ program is a collection of sub-programs called **"Functions"**.

❖ A Function is a block of code.

❖ A C++ program should have a function called **main()** function.

❖ Program execution starts by the **main()** function.



❖ C++ Constructs (Vocabularies/ Alphabets):

1 CHARACTERS

A …. Z a …. z	Letters
0, 1, 2 …. 9	Digits
SPACE TAB ENTER	Whitespaces or Blanks

2 SPECIAL SYMBOLS

{ } ( ) < > [ ] + - * / % . ; , ! ? & | = # “ ‘ …

Some symbols are not allowed, for example: ÷ @ ≤ ∑

3 RESERVED WORDS (OR KEYWORDS)

- Keywords are words with special meanings.
- C++ has more than 70 keywords like (*keywords are all in lowercase letters*):

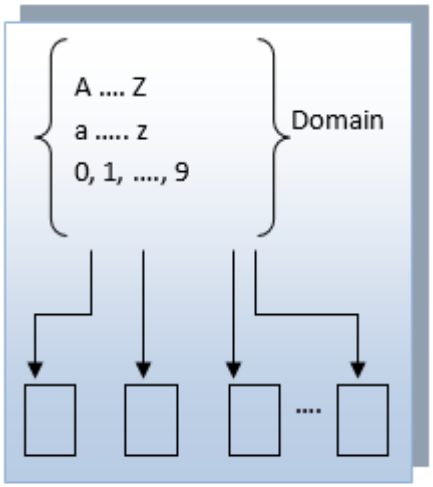
```
auto   const     double  float  int       short
struct   unsigned   break  continue  else    for
long      signed  switch   void  case   default   enum
goto   register  sizeof  typedef  volatile
char   do        extern  if     return    static
union    while
```

❖ C++ Token:

- A ***token*** is the smallest individual unit in a program.
- For Example: + is a token, ++ is a token, and == is a token.
- C++ tokens involve:
  - Reserved words.
  - Special Symbols.
  - Identifiers.

❖ Identifiers:

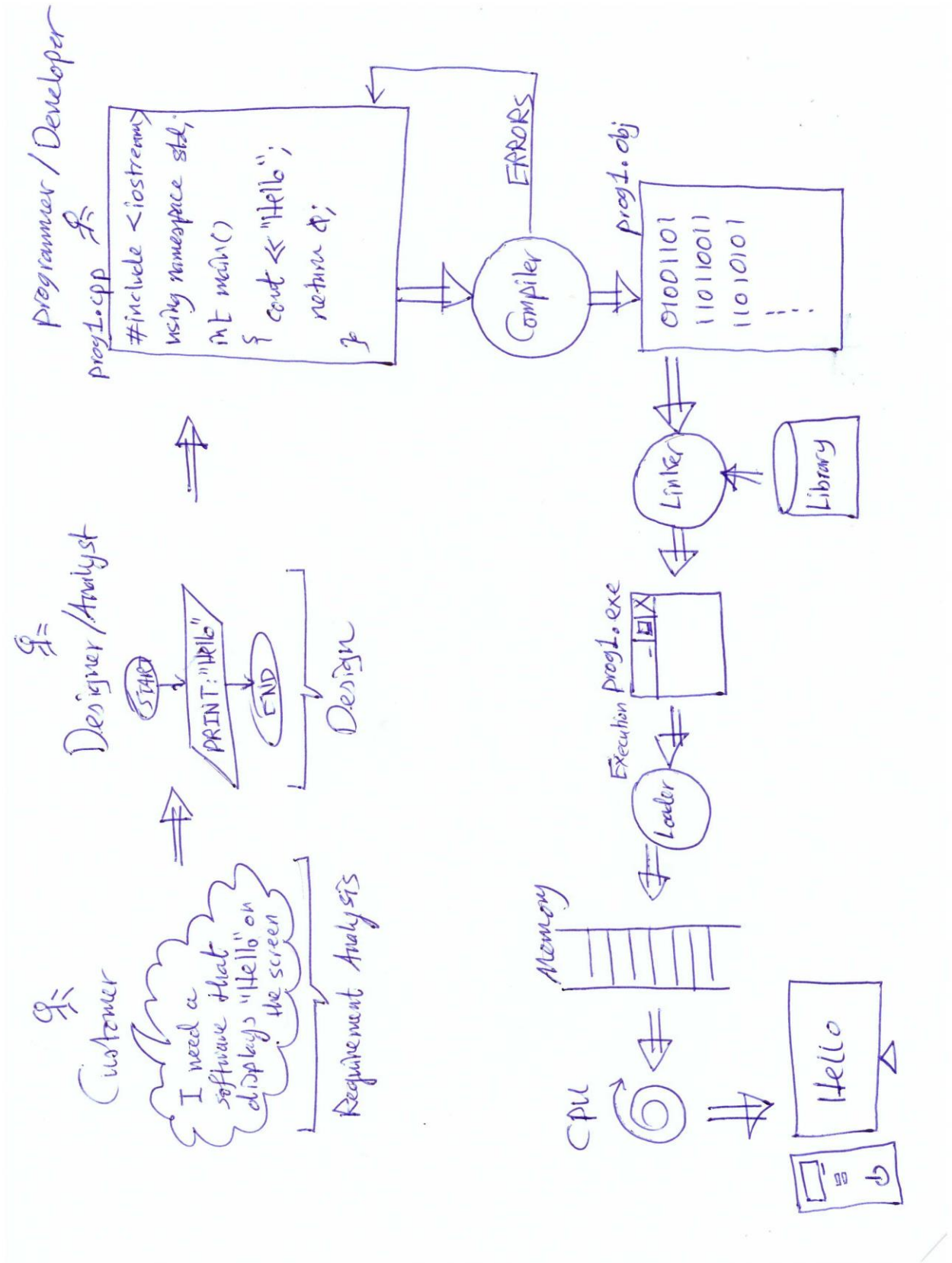
- Identifiers are rules of creating valid names in a C++ program.
- Names includes the names of variables, constants, functions, ….
- These rules are:
  - C++ identifier consists of letters, digits and the underscore character _ (cannot contain a special symbol.)
  - C++ identifier cannot begin by a digit.
  - C++ identifier cannot be a reserved word.



Exercise: ***Determine whether the following C++ identifiers are Valid or Invalid?***

	Identifier	VALID/ INVALID		Identifier	VALID/ INVALID
[1]	Sum_of_Squares		[9]	Total	
[2]	G 10		[10]	40Hours	
[3]	Box_22		[11]	_Count	
[4]	alpha-12		[12]	object(1)	
[5]	else		[13]	age#1	
[6]	Return		[14]	Folder2000	
[7]	one+two		[15]	2ndValue	
[8]	my salary		[16]	var.123	

Problem Analysis-Coding-Execution Cycle



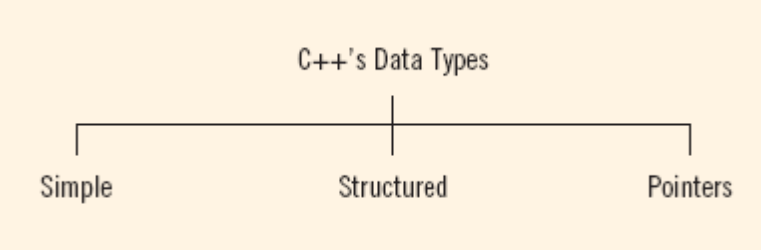
My First C++ Program

❖ Any computer program can be viewed as:

- Data
- Operations (or instructions) to manipulate these data

C++ Data Types

- A Data Type: set of values together with set of operations.
- C++ Supports the following categories of data types:



- Simple data types include: (handled as one unit in memory)
  - Integral
  - Floating-Point
  - Enumeration

- **Integral Data Type:**
  - Includes: **int**, **char**, **bool**

1 The **int** data type

- Used to hold whole numbers (numbers without fractional part).
- Example: 12, 0, -658, +98, 67, 35871 are all integer numbers.
- No commas are used within an integer.
  For instance: **12,638** is an invalid integer value in C++.

2 The **bool** data type

- Used to hold logical values.

true false

- **true**, **false**, and **bool** are all reserved words.

3 The **char** data type

- Used to hold character data (alphanumeric values).
- Characters include:
  - Letters (A to Z and a to z)
  - Digits (0, 1, 2, …. , 9)
  - Special Symbols (#, !, @, &, %, ….)

- A character value is enclosed between two single quotes ‘ ’
  ‘A’ ‘b’ ‘5’ ‘+’ ‘>’ ‘;’ ‘ ’

- ‘abc’ ‘520’ ‘v1’ ‘Hello!’ are all invalid characters, why?

- C++ uses ASCII code to represent characters. Each character has a value.

‘0’ → 48	‘A’ → 65	‘a’ → 97
‘1’ → 49	‘B’ → 66	‘b’ → 98
⋮	⋮	⋮
‘9’ → 57	‘Z’ → 90	‘z’ → 122

By the way:
ASCII stands for American Standard Code for Information Interchange.

- Why ASCII?
- How many bits used to encode each character in ASCII? _____
- How many characters are represented in ASCII? _____
- Is there any other character representation code?

ASCII CODE

ASCII control characters

00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters

32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Extended ASCII characters

128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	█	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

❖ Floating-Point Data Type:

- Includes: **float**, **double**
- Used to hold real numbers (numbers with fractional part).
- Floating-point numbers can be represented in two notations:
  - Decimal : Ex: 5400.0 0.0001
  - Scientific: Ex: 5.4E+003, 1.0E-004

- Examples:

$$1.2345 = \underbrace{12345}_{\text{significand}} \times \underbrace{10^{\overbrace{-4}^{\text{exponent}}}}_{\text{base}}$$

Real Number	C++ Floating-Point Notation
75.924	7.592400E1
0.18	1.800000E-1
0.0000453	4.530000E-5
-1.482	-1.482000E0
7800.0	7.800000E3

By the way:

.25 and 25. are valid C++ Floating-Point numbers.

.25 ≡ 0.25

25. ≡ 25.0

1 **float** data type

- The data type float is used in C++ to represent any real number between -3.4E+38 and 3.4E+38.

2 **double** data type

- The data type double is used in C++ to represent any real number between -1.7E+308 and 1.7E+308.

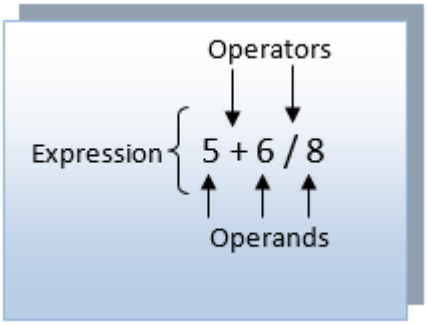
Note: Minimum and Maximum values are system dependent (compiler dependent).

The string Data Type

- The string data type is not a built-in C++ data type.
- Requires #include <string>
- The string data type is used to store text values.
- A C++ string is enclosed between two double quotations **"……."**
- A string is sequence of zero or more characters.
  - For Example: "Hello World" "I am 18 years old!" "A" "2017"
- Zero-length string is called the NULL string **""**;
- C++ string cannot be separated into multiple lines.

Example: "I Live
in Palestine." } **Invalid**

C++ Arithmetic Operators



- C++ is capable of performing Addition (+), Subtraction (-), Multiplication (*), Division (/), and Modulus (or Remainder) (%).
- Arithmetic expression: numbers separated by arithmetic operators.
- Data at which the operator to be applied on are called: Operands.
- An operator can be:
  - Unary Operator: has a single operand. (Ex: +5, -num)
  - Binary Operator: has two operands. (Ex: , 15 + 6, x * y)
- **+** and **-** are unary and binary operators.
- *****, /, **%** are binary operators. (6 * is not a valid expression)

❖ The Division Operator /

- If the two operands are integers, the result is an integer.
- Examples: 5 / 2 ➔ 2 4 / 9 ➔ 0 14 / 3 ➔ 4 7 / 2 ➔ 3
  7.0 / 2 ➔ 3.5 7 / 2.0 ➔ 3.5 7.0 / 2.0 ➔ 3.5
  5 / 0 ➔ Logical ERROR

❖ The Modulus Operator %

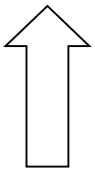
- % is used only with integer operands, it yields the remainder of division.
- Examples: 34 % 5 ➔ 4 15 % 4 ➔ 3 2 % 7 ➔ 2
  25.5 % 3 ➔ ERROR 39 % 0 ➔ Logical ERROR

Exercise: *What is the output of the following C++ Program?*

```
#include <iostream>
using namespace std;
void main()
{
    cout << "2 + 5 = " << 2 + 5 << endl;
    cout << "3.0 + 9.4 = " << 3.0 + 9.4 << endl;
    cout << "34 - 20 = " << 34 - 20 << endl;
    cout << "16.3 - 5.2 = " << 16.3 - 5.2 << endl;
    cout << "2 * 7 = " << 2 * 7 << endl;
    cout << "5 / 2 = " << 5 / 2 << endl;
    cout << "4.2 * 2.5 = " << 4.2 * 2.5 << endl;
    cout << "37 % 5 = " << 37 % 5 << endl;
    cout << "4 % 6 = " << 4 % 6 << endl;
}
```

❖ Order of Precedence:

- An expression is evaluated according to precedence rules.
- Associativity determines how operators of the same precedence are grouped in the absence of parentheses.

Precedence	Operator	Associativity
Higher  Lower	Parenthesis ( )	
	* / %	Left to Right
	+ -	Left to Right

❖ Examples:

```
cout << 2 + 3 * 5;
```

```
cout << (2 + 3) * 5;
```

```
cout << 7 / 2 * 3;
```

```
cout << 39 % 4 * 3 / 5;
```

```
cout << 6 / 4 + 3.9;
```

```
cout << 'A' + 2;
```

```
cout << 3 * 7 - 6 + 9 * 5 / 4 + 5;
```

```
cout << "Hello"
     << "World";
```

```
cout << 9 + 7 * 5  % 3 - 5 * 9 % 11 - 8 << endl;
```

EXAMPLE:

```
#include <iostream>
using namespace std;
int main()
{
     cout << "3 / 2 + 5.5 = " << 3 / 2 + 5.5 << endl;
     cout << "15.6 / 2 + 5 = " << 15.6 / 2 + 5 << endl;
     cout << "4 + 5 / 2.0 = " << 4 + 5 / 2.0 << endl;
     cout << "4 * 3 + 7 / 5 - 25.5 = "
     << 4 * 3 + 7 / 5 - 25.5 << endl;
     return 0;
}
```

C++ COMMENTS

- Comments are explanatory Sentences in the source code.
- Comments are not executed, and are ignored by the compiler.
- Comments are directed to the reader of the program.
- There are two styles of writing comments in C++:

C++ Style Single-Line Comments

```
// ……..
// ……..
```

C Style Multiple-Line Comments

```
/*
……..
……..
*/
```

HOW C++ ALLOCATES MEMORY LOCATIONS?

Using **Declaration Statement** instructs the computer to put data into the computer's memory.

❖ **SYNTAX of Declaration Statement:**

dataType identifier; OR dataType identifier, identifier, identifier, ….;

Variable Name

`int` `char` `bool` `double` `float` `string`

❖ A Declaration Statement reserves a memory location called ***variable***.

❖ The content (value) of a ***variable*** may change during program execution.

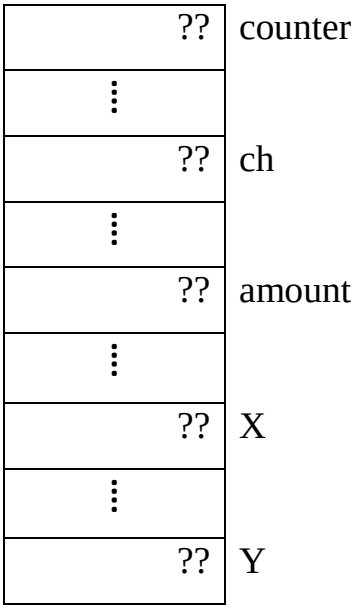
❖ Examples:

By the way:

- **A variable is a memory location.**
- **NO variables may have the same names (in the same block).**

```
int  counter;
/* tells the computer to allocate a memory location to store an integer value */

char ch;
// ch is a variable, it will hold a character value

double  amount;
// amount will contain a real number
cout << amount;  // ?? UNKNOWN

int  X, Y;
// two integer variables are created
```

Memory

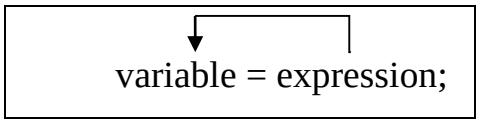


EXERCISE: *Which of the following C++ declaration statements are correct?*

Declaration		VALID/ INVALID	Declaration		VALID/ INVALID
[1]	string my name;		[6]	int var@2017;	
[2]	Bool isOK,		[7]	char c1, c2, c3;	
[3]	char ch#1;		[8]	int Double;	
[4]	int I, j, k;		[9]	int 10;	
[5]	double 4sale;		[10]	int return;	

❖ Assignment Statement:

- Assigns (stores) a value into a variable. (= is called the assignment operator)
- **SYNTAX of Assignment Statement:**

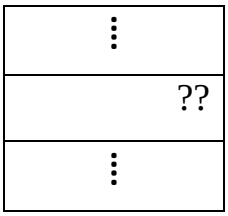


EXAMPLE:

int num; // declares an integer variable called ***num***

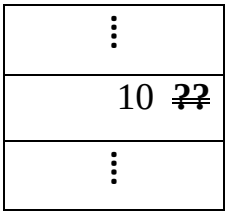
cout << num;

// UNKOWN value will be printed



num = 10; //Assignment

cout << num;



num = 8 * 5 – 13;

cout << num; **27**

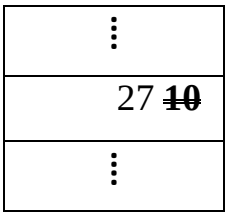
2 + 3 = num; // ERROR

38 = num; // ERROR

num = num + 2; // Valid

cout << num; // Prints 29

Memory



EXAMPLE:

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
      int  val;  char ch;
      float tax; string str;
```

cout << "val"; ----> 

cout << val; ------> 

val = 6 + 4 * 2;

cout << val; ------> 

val = 17 % 7 ;

cout << val; ------> 

ch = 'A';

cout << ch; ------> 

ch = A; //ERROR

ch = '@'; // VALID

ch = '7'; // VALID

int quantity = 12,000; //ERROR

ch = '465'; // ERROR

val = 465; // VALID

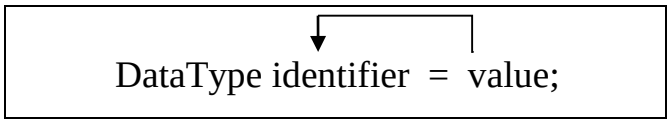
tax = 8.15;
cout << tax; ------> 

tax = 1.2E3; // VALID
cout << tax; ------> 

```
tax = 5.0 / 2;  cout << tax;   // Prints ______

str =  I'm Programming in C++;   // ERROR
str =  "I'm Programming in C++";   // VALID
cout << "str";  // Prints _______
cout << str;     // Prints _______

double ch;  // ERROR, NO variables may have the same name
double Ch;  // VALID

return 0;
}
```

❖ Initialization Statement:

- Assigning a value into a variable ***during declaration***.
- **SYNTAX of an Assignment Statement:**



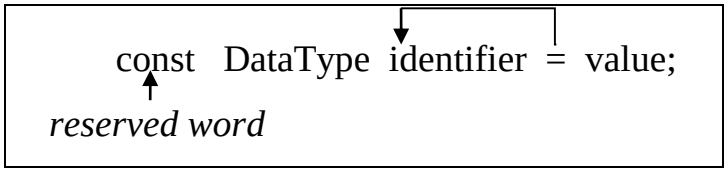
EXAMPLES:

```
int  num = 5;
char code = 'm';
string text = "return";
float alpha = .83;
int x = 10, y = 15;
double beta = ;  // ERROR
```

```
int a = 1, b, c=2;
b = c;  // VALID
cout << c; // prints ______
int temp = 12 – 4  % 3;    // VALID
int nVar = a * b – c;       // VALID
```

❖ Named Constant:

- A named constant is a named memory location at which its value (i.e. content) ***cannot be changed***.
- **SYNTAX of a Named Constant:**



EXAMPLE:

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
     const float PI = 3.14;
     const double TAX = 0.17;
     const char STAR = '*';
     const int MAX = 20;
     const int MIN;   //ERROR

     PI = 3.1428;   //ERROR
     TAX = 0.15;  //ERROR

     int const;     //ERROR
     string STAR;   //ERROR
     const string PPU = "Palestine Polytechnic University";
     cout << PPU;   // Prints ____________________________
      ⋮
}
```

The Input (Read) Statement

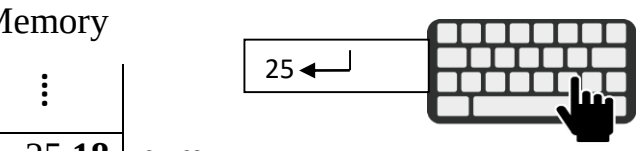
- A C++ program may need to input data ***during execution***.
- The ***cin*** and the extraction operator >> are used to prompt the user to enter a value (*in most program environments from the keyboard*), and to put it into a variable located in memory. (Interactive input)

- **SYNTAX of C++ read statement:**

cin >> variable; **OR** **cin >> variable >> variable >> ….;**

EXAMPLE:

```
int  num = 18;
cout << num;          // prints _____

cin >> num;
// The program pauses asking the user to
// enter a value from the Keyboard

cout << num;          // prints _____
cin >> num * 2;  // ERROR
cin >> 68;        // ERROR

const int AMOUNT = 100;
⋮
cin >> AMOUNT;   // ERROR, why?
```



By the way:

<< **is called the Insertion Operator.**

>> **is called the Extraction Operator.**

EXAMPLE:

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
     string fstName;
     string LstName;
     int age;
     double weight;

     cout << "Enter first name, last name, age, "
     << "and weight:" << endl;

     cin >> fstName >> LstName;
     cin >> age >> weight;
     cout << "Name: " << fstName << " " << LstName << endl;
     cout << "Age: " << age << endl;
     cout << "Weight: " << weight << endl;
     return 0;
}
```

EXERCISE:

Write a C++ program that reads the radius of a circle, then finds and displays the area and circumference of this circle. (Draw the flowchart first).

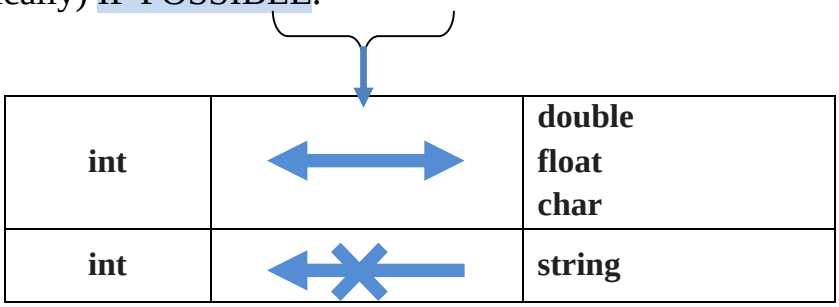
Hint: ***area*** $= \pi r^2$, ***circumference*** $= 2\pi r$

```
// This program will find the area and circumference of a circle
#include <iostream>
using namespace std;
int main()
{




}
```

Type Conversion (Casting and Coercion)

1 Coercion:

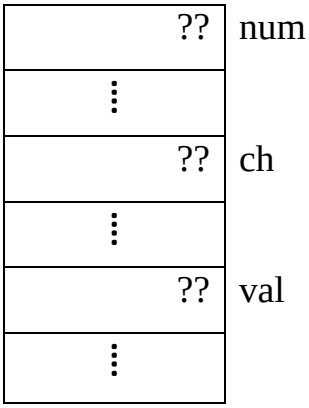
- Coercion means converting a **value** from data type to another data type **implicitly** (automatically) IF POSSIBLE.



EXAMPLE:

```
int  num;
char ch;
float val;

double num;  // ERROR
num = 4.9;  // Coercion
cout << num;  // Prints _____

val = 2;  // Coercion
cout << showpoint;
cout << val;  // Prints _____

ch = 65;  // Coercion
cout << ch;   // Prints _____


val = "Hello";   //ERROR
num = "ABCD";  // ERROR
num = "321";  // ERROR


num = 'B';  // Coercion
cout << num;  // Prints _____
```

Memory



Note:

C++ will not print the fractional part if it equals ZERO.

```
float f = 8.0;
cout << f;
```

8

```
// showpoint forces the fraction to be
// printed even if it equals ZERO.
float f = 8.0;
cout << showpoint;
cout << f;
```

8.0

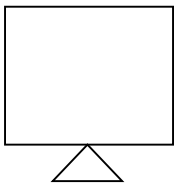
2 Casting:

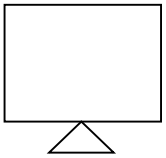
- Casting means converting a **value** from data type to another data type **explicitly** IF POSSIBLE.
- C++ provides a cast operator **static_cast** to perform casting:

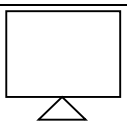
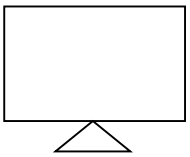
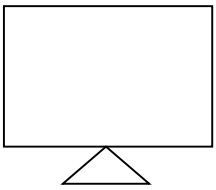
static_cast<DataType> (value);

- **static_cast** is a keyword.

EXAMPLES:

1	`cout << static_cast<int>(7.9) << endl;`
	
	`cout << showpoint;`
	`cout << static_cast<double>(25) ;`
		

2	`cout << static_cast<int> ('A');`
	
	`cout << static_cast<char> (100);`
		

3	`cout << static_cast< float > (15) / 2;`	
4	`cout << showpoint;` `cout << static_cast<float> (15/2);`	
5	`cout<< static_cast<int>( 7.8 + static_cast<double>(15) / 2 );`	

NOTE: Casting will not change the variable's data type.

EXAMPLE:

```
int alpha;
double beta;

beta = 2.5;
alpha = static_cast<int> (beta);  //  or alpha = beta;

cout << alpha;    // Prints _________
cout << beta;     // Prints __________
```

EXERCISE:	*What is the output of the following C++ code?*

Code Fragment	OUTPUT
int a = 5, b = 10; **char ch;** **cin >> a;** **cin >> ch;** **cin >> b;** } **cin >> a >> ch >> b;** **cout << a << endl << b << endl << ch;**	**Assume the input is:** **27 R –18** ↵

```
int firstNum, secondNum;
double z;
char ch;
string name;

firstNum = 4;
secondNum = 2 *firstNum + 7;
z = (firstNum + 1) / 2.0;

cout << z << endl;

ch = 'A';
cout << secondNum << endl;

cin >> secondNum;
cin >> z;

firstNum = 2 *secondNum +
static_cast<int>(z);
cout << firstNum << endl;

cin >> name;
secondNum = secondNum + 1;
cin >> ch;   cout << ch << endl;

// 'M' in ASCII  = 77
firstNum = firstNum + static_cast<int>(ch);
z = firstNum - z;

cout << z << endl << firstNum << endl <<
secondNum;
```

Assume the input is:
8 15.3 Jenny M↵

Increment and Decrement Operators

- C++ provides the increment operator ++, which increases the value of a variable by 1, and the decrement operator, ––, which decreases the value of a variable by 1.

Pre-increment: ++variable
Post-increment: variable++
Pre-decrement: ––variable
Post-decrement: variable––

var++ ; and ++var; ≡ var = var +1;

var-- ; and --var; ≡ var = var -1;

- Example:

```
int ctr = 5;
ctr++;  // or  ++ctr;
cout << ctr; // Prints 6
(ctr+2)++; // ERROR
58++; // ERROR

double beta = 9.65;
beta --;  // or  -- beta;
cout << beta; // Prints 8.65

char ch = 'c';
ch++;
cout << ch; // Prints _____
```

- pre- and post- operators are different from each other when these operators are applied within an expression.
- For instance:

N = var++;	1) N = var; 2) var = var + 1;
N = ++var;	1) var = var +1; 2) N = var;

- Example:

int v = 8;
cout << v++; // Prints _____
cout << v; // Prints _____
	int v = 8;
	cout << ++v; // Prints _____
	cout << v; // Prints _____

- Example:

```
int  a = 4, b = 7, c;

c = a++ * --b;

cout <<  a;   // Prints _____
cout <<  b;  // Prints _____
cout <<  c;  // Prints _____
cout << ++c;  // Prints _____

a = b++ * 7;
cout <<  a;   // Prints _____
cout <<  b;  // Prints _____

c = 2 * a + ++b;
cout <<  c;  // Prints _____
cout <<  b;  // Prints _____
```

Escape Sequences

- Escape sequences are special characters that have special meanings inside a string or a character.
- C++ Escape Sequences: **\n \t \' \" **

  Consider the following table for their meanings:

	Meaning	
'\n'	New Line	↵
'\t'	Tab	SPACES
'\\'	Backslash	\
'\"'	Double Quotation	"
'\''	Single Quotation	'

Remember:

```
char ch;
ch = 65;  // Legal, coercion
ch = '65';  // ERROR

cout << "I am Studying
        a C++ Course.";  // not Legal
```

EXAMPLES:

1	`cout << "Hello \nWorld";`
	`cout << "Hello" << '\n' << "World";`
	`cout << "Hello" << endl << "World";`	

2	`cout << "I Live in "Palestine" country."; //ERROR`
	`cout << "I Live in \"Palestine\" country.";`	

3	`cout << "\\C++\\ Course";`	
4	`cout << "Hello \t World";`	
5	`cout << "Hello \\t World";`	

6	`cout << "Hello \nthere. \nMy name is James." << endl;`
	`cout << "The newline escape sequence is \\n" << endl;`
	`cout << "The tab character is \'\\t\'" << endl;`	

7	`cout << "String \"Sunny\" contains five characters.";`	

More on Assignment Statements

- C++ provides compound assignment operators: **+=** **-=** ***=** **/=** **%=**
- If a and b are two variables, then:

  a += b; ➔ a = a + b;

  a -= b; ➔ a = a - b;

  a *= b; ➔ a = a * b;

  a /= b; ➔ a = a / b;

  a %= b; ➔ a = a % b;

- Example:

  int x = 5, y = 7;

  x += y; // x = x + y;

  cout << x; // Prints 12

EXAMPLE 2-31

This example shows several compound assignment statements that are equivalent to simple assignment statements.

Simple Assignment Statement	Compound Assignment Statement
`i = i + 5;`	`i += 5;`
`counter = counter + 1;`	`counter += 1;`
`sum = sum + number;`	`sum += number;`
`amount = amount * (interest + 1);`	`amount *= interest + 1;`
`x = x / ( y + 5);`	`x /= y + 5;`

EXERCISES:

[1] *Use a compound assignment operator to convert the following simple assignment:*

x = x * y + z – 5;

[2] *Write the equivalent C++ code for each of the following mathematical equations:*

$$c = \frac{5}{9}(f - 32)$$

$$A = \frac{B+C}{D-E}$$

$$root = \frac{-b+(b^2-4ac)}{2a}$$

[3] *Write a C++ program that reads a 3-digit number, and prints its digits on separate lines.*

[4] *Newton's law states that the force, F, between two bodies of masses M1 and M2 is given by:*

$$F = k\left(\frac{M_1 M_2}{d^2}\right)$$

in which k is the gravitational constant and d is the distance between the bodies. The value of k is 6.67 X 10^{-8} dyn.cm²/g². Write a C++ program that prompts the user to input the masses of the bodies, and the distance between the bodies. The program then outputs the force between the bodies.

THE END