PROG0101
FUNDAMENTALS OF PROGRAMMING

Chapter 1

Overview of Computers and Logic

Topics

- Definition of Computer
- Computer Operations
- Machine Cycle
- Programming Terminology
- The Programming Process

Definition of Computer

- Definition of computer:
  - a computer is a programmable machine.

Definition of Computer

- What computers can do?
  - Office works (documents, letters, memos etc …)
  - Entertainment
  - Access information
  - Education
  - Online shopping
  - Video conferencing
  - Desktop publishing
  - Etc …

Computer Parts

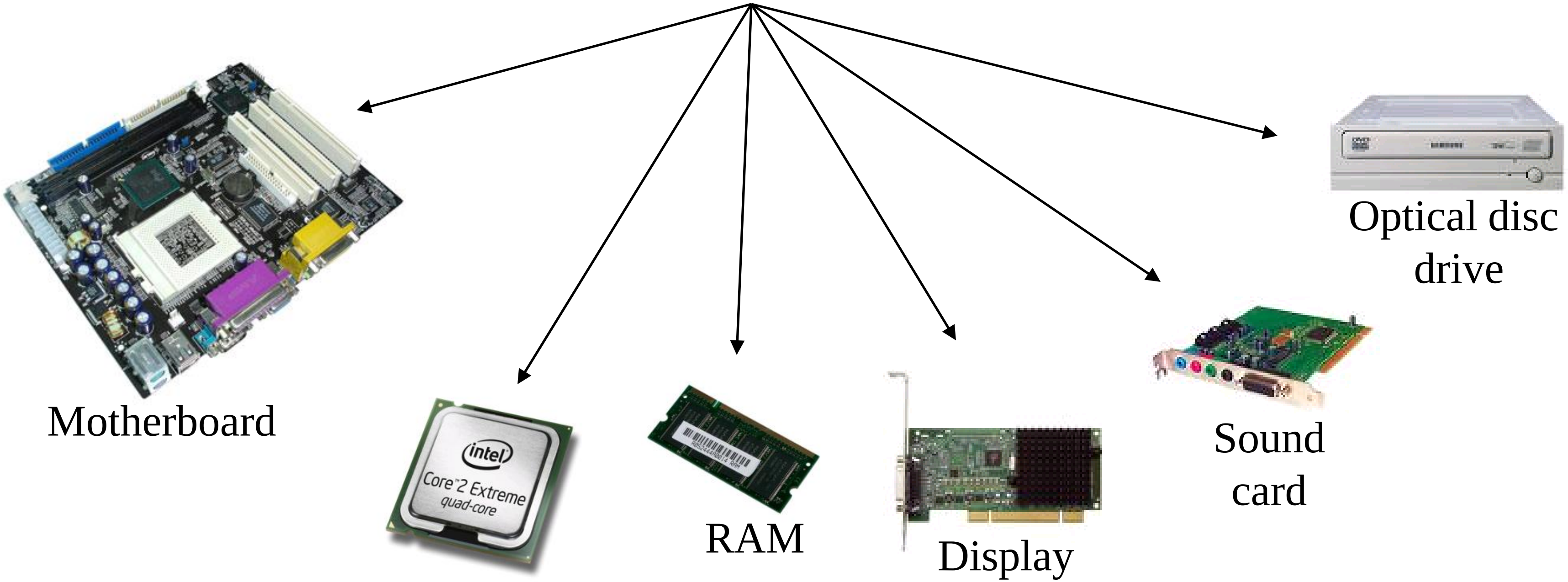
- Computer can be divided into two parts:
  - Hardware
    - System Unit
    - Input devices
    - Output devices
  - Software
    - System softwares
    - Programming softwares
    - Application softwares

Computer Parts - Hardware

- The **system unit** is the core of a computer system.
- Usually it's a rectangular box placed on or underneath your desk.
- The most important components are
  - Central processing unit (CPU)
  - Random access memory (RAM)
  - Hard disk
  - Etc.
- Hardware that is not part of the system unit is sometimes called a peripheral device or device.

Computer Parts - Hardware

System Unit



Motherboard

Processor

RAM

Display card

Sound card

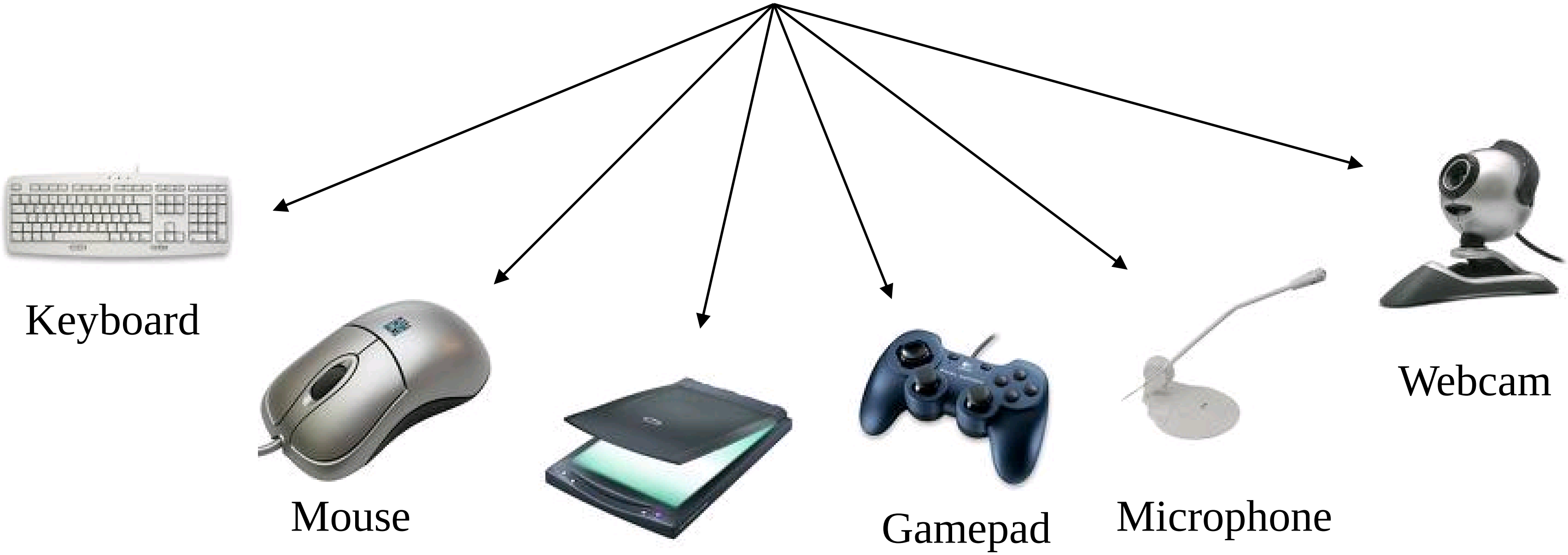
Optical disc drive

Computer Parts - Hardware

- An **input device** is any hardware device that sends data to the computer.
- Without any input devices, a computer would only be a display device and not allow users to interact with it,

Computer Parts - Hardware

Input Devices



Keyboard

Mouse

Scanner

Gamepad

Microphone

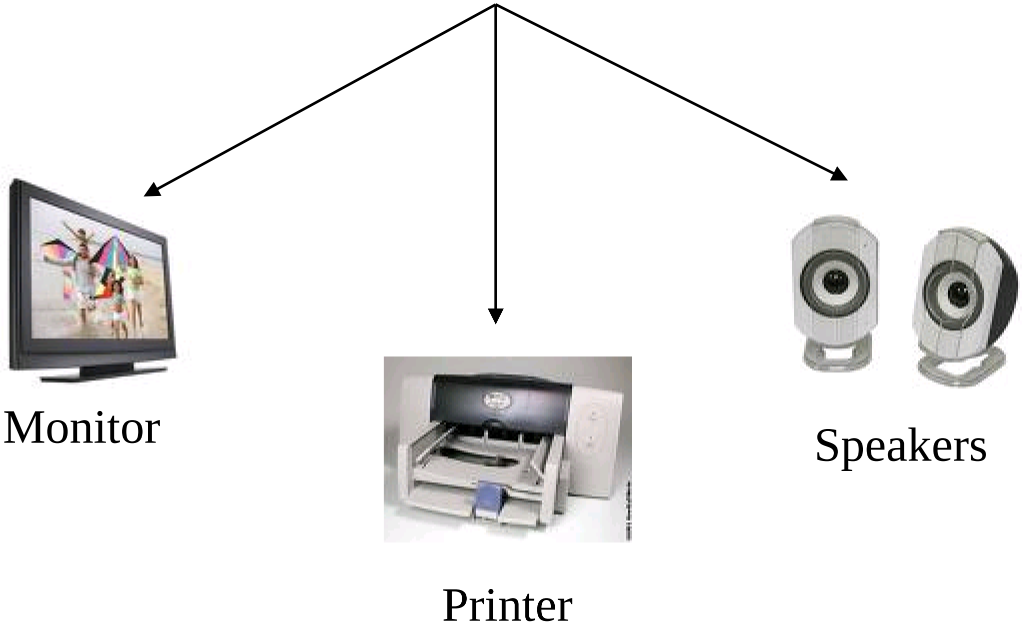
Webcam

Computer Parts - Hardware

- An **output device** is any piece of computer hardware equipment used to communicate the results of data processing carried out by a computer to the outside world.

Computer System - Hardware

Output Devices



Monitor

Printer

Speakers

Computer System - Software

System Software

- System software is computer software designed to operate the computer hardware and to provide and maintain a platform for running application software.

Computer System - Software

System Software - Example

- Operating System
- BIOS
- Utility software

Computer System - Software

Programming Software

- A programming tool or software development tool is a program or application that software developers use to create, debug, maintain, or otherwise support other programs and applications.
- **Integrated Development Environments (IDE)** combine the features of many tools into one package.
- They for example make it easier to do specific tasks, such as searching for content only in files in a particular project.

Computer System - Software

Programming Software

- An IDE normally consists of:
  - a source code editor
  - build automation tools
  - a debugger

Computer System - Software

Programming Software - Example

- Microsoft Visual Studio
- Eclipse
- JBuilder
- Komodo

Computer System - Software

Application Software

Application software is computer software designed to help the user to perform singular or multiple related specific tasks.

Computer System - Software

Application Software - Example

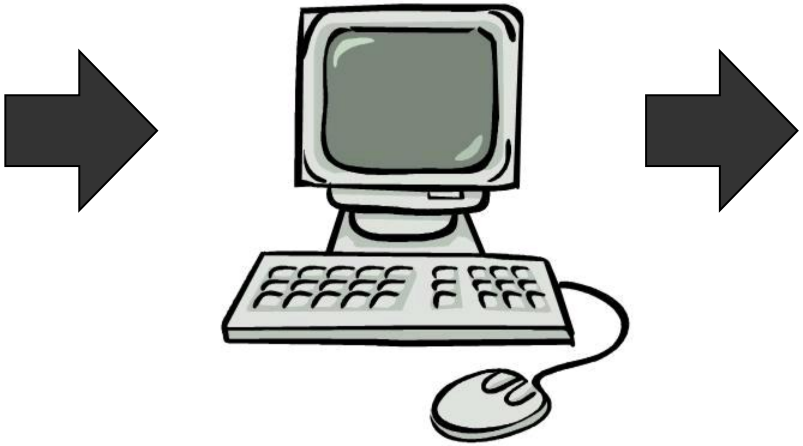
- Word processor
- Spreadsheet
- Desktop publishing
- DBMS
- Video editor
- Web browser

Basic Computer Operations

The computer performs four basic operations:

- Receive input data
- Process data
- Produce output information
- Store data, information and instructions.

Basic Computer Operations



PROCESS DATA

DATA
(INPUT)

INFORMATION
(OUTPUT)

STORE
DATA, INFORMATION &
INSTRUCTION

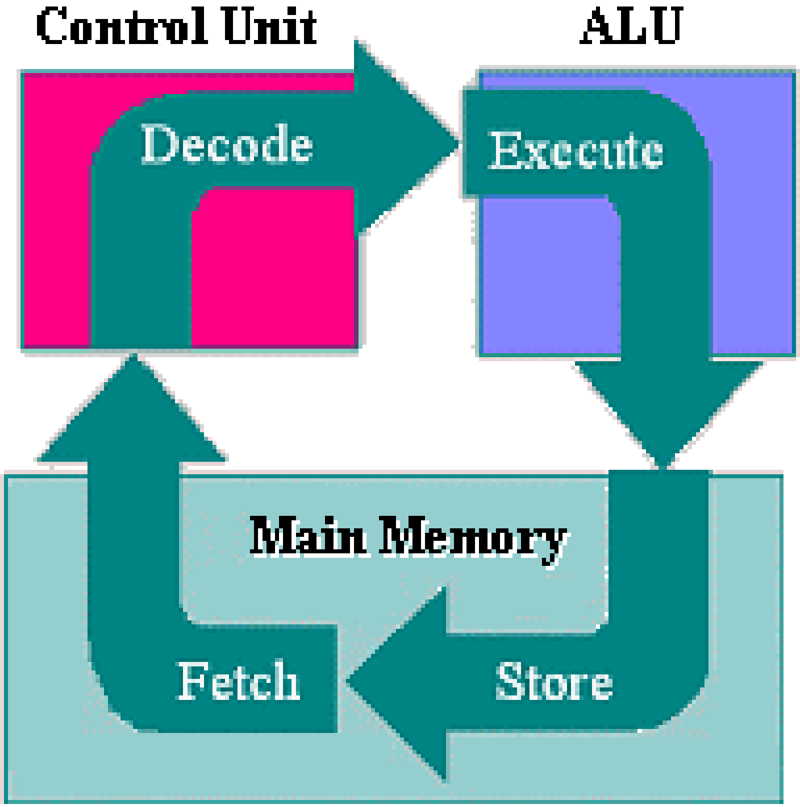
Machine Cycle

- Machine cycle is the basic operation cycle of a computer processor.
- It is the process by which a computer processor retrieves a program instruction from its memory, determines what actions the instruction requires, and carries out those actions.
- This cycle is repeated continuously from bootup to when the computer is shut down.

Machine Cycle

- There are four steps in machine cycle:
  - **Instruction Phase**:
    - Fetch - get an instruction from Main Memory
    - Decode - translate it into computer commands
  - **Execution Phase**
    - Execute - actually process the command
    - Store - write the result to Main Memory

Machine Cycle



Machine Cycle

Example

For example, to add the numbers 5 and 6 and show the answer on the screen requires the following steps:

1. Fetch instruction: "Get number at address 123456"
2. Decode instruction.
3. Execute: ALU finds the number. *(which happens to be 5)*
4. Store: The number 5 is stored in a temporary spot in Main
Memory.

5 - 8 Repeat steps for another number (= 6)

9. Fetch instruction: "Add those two numbers"
10. Decode instruction.
11. Execute: ALU adds the numbers.
12. Store: The answer is stored in a temporary spot.
13. Fetch instruction: "Display answer on screen."
14. Decode instruction.
15. Execute: Display answer on screen.

Common Features of All Program

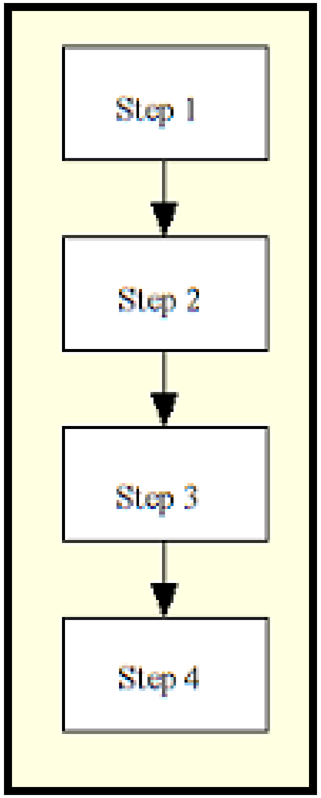
All programs could be structured in the following four ways:

- Sequences of instructions
- Branches
- Loops
- Modules

Common Features of All Program

Sequences of Instructions

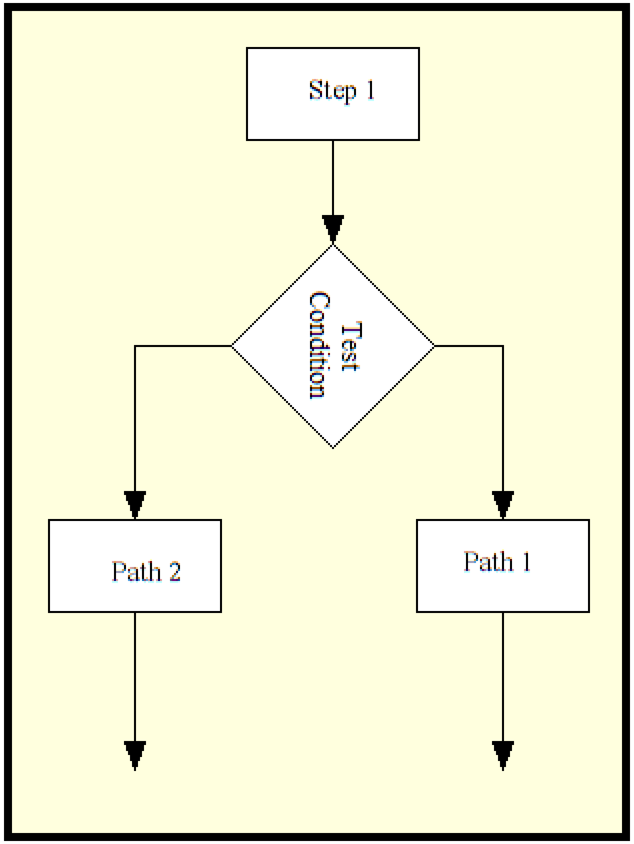
The program flows from one step to the next in strict sequence.



Common Features of All Program

Branches

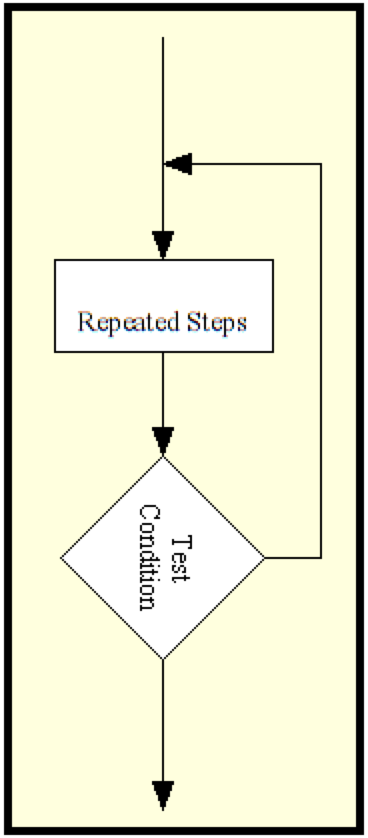
The program reaches a decision point and if the result of the test is true then the program performs the instructions in Path 1, and if false it performs the actions in Path 2



Common Features of All Program

Loops

The program steps are repeated continuously until some test condition is reached, at which point control then flows past the loop into the next piece of program logic



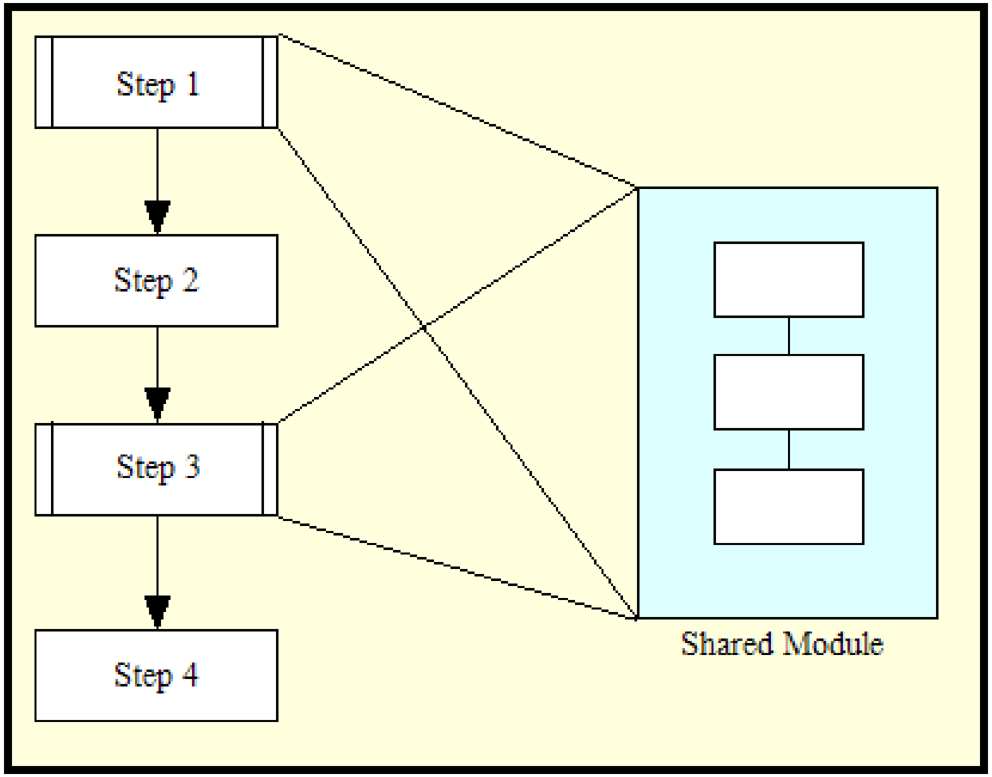
Common Features of All Program

Modules

The program performs an identical sequence of actions several times. For convenience these common actions are placed in a module, which is a kind of mini-program which can be executed from within the main program.

Common Features of All Program

Modules



Common Features of All Program

Along with these structures programs also need a few more features to make them useful:

- Data (we take a closer look at data in the Raw Materials topic)
- Operations (add, subtract, compare etc)
- Input/Output capability (e.g. to display results)

The Programming Process

Step 1: Defining the problem.

Step 2: Planning the solution.

Step 3: Code the program.

Step 4: Test the program.

Step 5: Document everything.

The Programming Process

Step 1: Defining the problem

- The task of defining the problem consists of identifying what it is you know (input-given data), and what it is you want to obtain (output-the result).
- Eventually, you produce a written agreement that, among other things, specifies the kind of input, processing, and output required.

The Programming Process

Step 1: Defining the problem

Example:

- What must the program do?
- What outputs are required and in what form?
- What inputs are available and in what form?

The Programming Process

Step 2: Planning the solution

- Two common ways of planning the solution to a problem are to draw a **flowchart** and to write **pseudocode**, or possibly both.
- A **flowchart** is a pictorial representation of a step-by-step solution to a problem.
- It consists of arrows representing the direction the program takes and boxes and other symbols representing actions.
- It is a map of what your program is going to do and how it is going to do it.

The Programming Process

Step 2: Planning the solution

- **Pseudocode** is an English-like nonstandard language that lets you state your solution with more precision than you can in plain English but with less precision than is required when using a formal programming language.
- Pseudocode permits you to focus on the program logic without having to be concerned just yet about the precise syntax of a particular programming language.

The Programming Process

Step 3: Code the program

- You will translate the logic from the flowchart or pseudocode-or some other tool-to a programming language.
- Program Coding means expressing the algorithm developed for solving a problem, in a programming language.

The Programming Process

Step 4: Test the program

- Almost all programs may contain a few errors, or bugs.
- Testing is necessary to find out if the program produces a correct result.
- Usually it is performed with sample data
- Debugging is the process of locating and removing errors

The Programming Process

Step 4: Test the program

Types of Error

Syntax Errors: Violation of syntactic rules in a Programming Language generates syntax errors.

Effect? Interpreter or Compiler finds it in Syntax Check Phase.

The Programming Process

Step 4: Test the program

Types of Error

Semantic Errors: Doing logical mistakes causes semantic errors in Source code.

Effect? Interpreters and Compilers can not notice them, but on execution, they causes unexpected results.

The Programming Process

Step 4: Test the program

Types of Error

Run-time Errors: Occur on program execution. Mostly caused by invalid data entry or tries to use not existing resources.

Effect? It occurs on run time and may crash the program execution

The Programming Process

Step 5: Document everything

- Documentation is a written detailed description of the programming cycle and specific facts about the program.
- Typical program documentation materials include the origin and nature of the problem, a brief narrative description of the program, logic tools such as flowcharts and pseudocode, data-record descriptions, program listings, and testing results.
- Comments in the program itself are also considered an essential part of documentation.

How Does a Computer Program Work?

- Computers don't do anything without someone telling them what to do.
- To make the computer do something useful, you must give it instructions in either of the following two ways:
    - **Write a program** that tells a computer what to do, step by step, much as you write out a recipe.
    - **Buy a program** that someone else has already written that tells the computer what to do.

How Does a Computer Program Work?

- A program does nothing more than tell the computer how to accept some type of input, manipulate that input, and spit it back out again in some form that humans find useful.

How Does a Computer Program Work?

Input and Output for Various Programs

Type of Program	Input	What the Program Does	Output
Word Processor	Characters you type from the keyboard	Formats the text; corrects spelling	Displays and prints neatly organized text
Game	Keystrokes or joystick movements	Calculates how fast and far to move a cartoon figure on-screen	Moves a cartoon figure on-screen
Stock-market predictor	Current and past prices for stocks	Tries to recognize trends in a stock's price fluctuations	Predicts the future price of a stock

How Does a Computer Program Work?

Input and Output for Various Programs

Type of Program	Input	What the Program Does	Output
Optical character recognition (OCR)	Text from a scanner	Recognizes shapes of characters	Converts scanned text into a text file that a word processor can edit
Web browser	HyperText Markup Language (HTML) codes on other computers	Converts the HTML codes into text and graphics	Displays Web pages on-screen