DEPARTMENT OF INFORMATION TECHNOLOGY
FACULTY OF MANAGEMENT STUDIES AND COMMERCE
UNIVERSITY OF SRI JAYEWARDENEPURA

# ITC 1370
# Information Technology for Business

## Introduction to Computer Programming

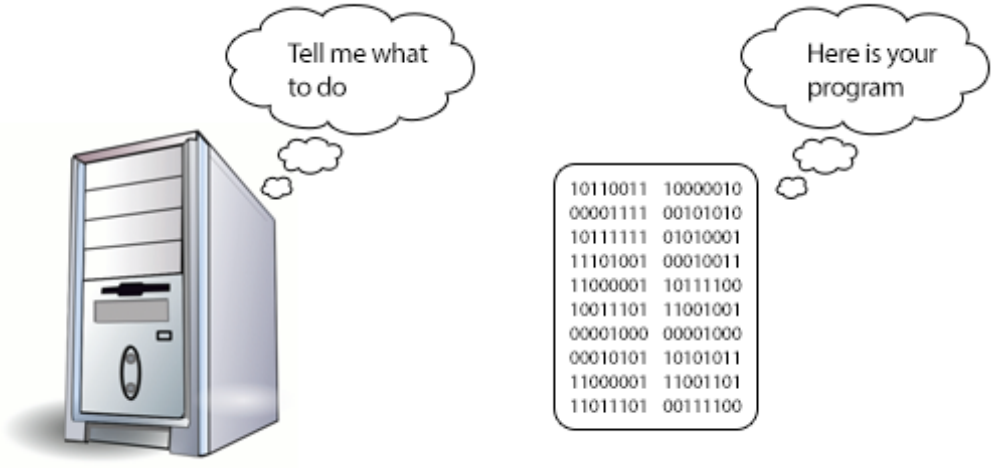
# What is a computer program?

A computer program is a **set of instructions** that a computer can follow to perform a specific task.

# What is computer programming?

Computer programming, also called coding, is all about **giving instructions to computers**.

It's like you're talking to a machine, but instead of natural language, you use a special language computers can understand. This language is full of precise instructions that tell the computer what tasks to complete and how to complete them.



# Computers can,

- **Perform Arithmetic and Logical Operations:** Computers can perform calculations incredibly quickly and precisely. This includes basic math operations like addition, subtraction, multiplication, calculus, and statistics. They can also perform logical operations, like comparing values (greater than, less than, equal to) and making decisions based on those comparisons.
- **Read/Display Data:** Computers can take in information from various sources, like keyboards, mice, scanners, and external drives. This information is then processed and displayed on the screen in a user-friendly format.
- **Write to Files:** Computers can store information on different storage devices like hard drives and solid-state drives. This allows them to save data for later use. Programs can write data to files in various formats, like documents, images, and videos.

# Program Development

Program development, also referred to as software development, is the entire process of creating a computer program.

It's like building a house - you don't just start putting up walls; there's planning, designing, and a lot more involved. Here's a main breakdown of the key stages in program development:

- Program Design
- Coding (Using a programming language)
- Program Testing

# *Program Design*

Here, you create a blueprint for the program. This involves outlining the program's structure, how different parts will interact, and what technologies will be used. It's like designing the floor plan and electrical wiring for your house.

Can use many methods to design a program.

- Use descriptions
- Use flowcharts
- Use UML Diagrams etc.

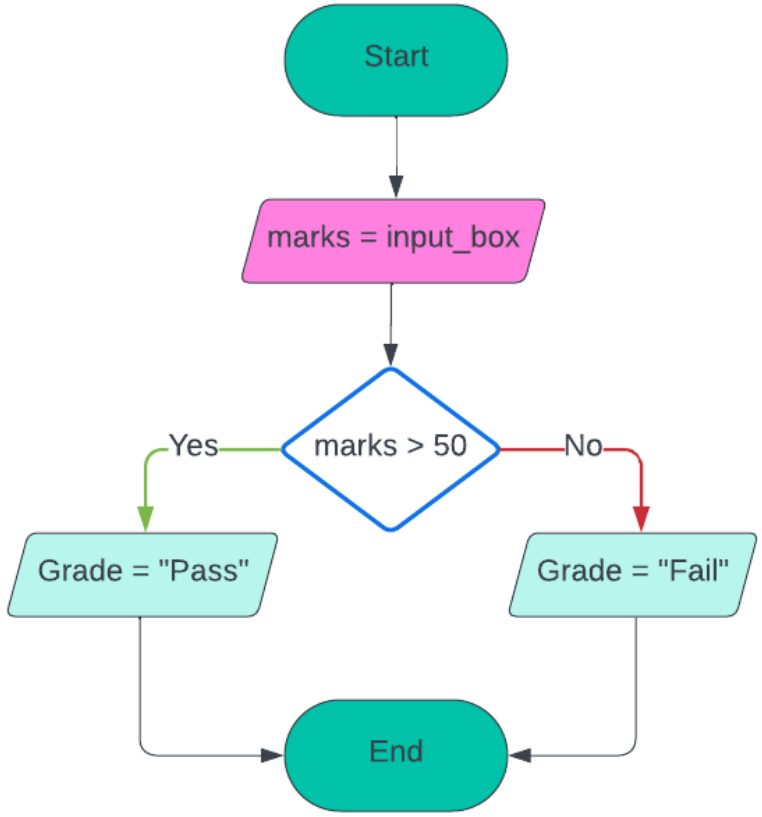
# Program Design (Example)

Program scenario: Enter subject marks. If marks are greater than 50, then Pass, else Fail.

**Use Descriptions**

The program will display an input box prompting the user to "Enter Subject Marks". The program will store the value entered by the user in a variable named "marks". The program will check if the value in "marks" is greater than 50. If marks are greater than 50, a message box will be displayed indicating "Pass". If marks are less than or equal to 50, a message box will be displayed indicating "Fail".

**Use Flow Charts**



# *Coding*

This is where the actual instructions, or code, are written in a specific programming language. It's like building the walls and putting in the plumbing based on the blueprint.

Different Programming Languages:

- **Machine Language** - Composed entirely of 1s and 0s (binary code) representing instructions for the processor. Incredibly difficult for humans to read, write, and understand due to its nature.
- **High level languages** - The most common type of programming languages, designed to be easier for humans to understand and write. Use keywords and syntax similar to natural languages (English, French, etc.).

  E.g.- JAVA, Python, C#, VBA

# *Program Testing (or Debugging)*

Once the code is written, it's rigorously tested to identify and fix any errors (bugs) that might prevent the program from working correctly. Imagine checking the electrical wiring and plumbing for any leaks before moving in.

E.g.- Based on the previous program scenario (in program design slide),

- Enter two numbers (higher than 50 and lower than 50) in the input boxes and see if the message box displays the correct result.
- Try entering non-numeric characters in the input boxes.

# Thank You