ITE 12 – FUNDAMENTALS OF PROGRAMMING

Lecture #1: Overview of Computer Programming and Problem Solving

OVERVIEW

- Basic computer concepts
- An overview of Computer Languages
- History of C programming language
- The program development life cycle
- Designing algorithm
- Flowcharting and pseudo-coding
- Introduction to C Programming

BASIC COMPUTER CONCEPTS

What is a Computer?

- Computer is an electronic device that accepts data from the user, processes it, produces results, displays them to the users, and stores the results for future usage

- Data is a collection of unorganized facts & figures and does not provide any further information regarding patterns, context, etc. Hence data means "unstructured facts and figures".

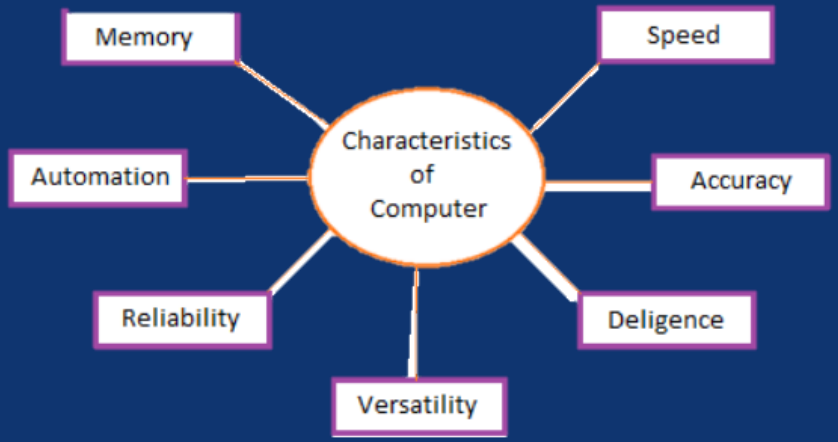
- Information is a structured data i.e. organized meaningful and processed data. To process the data and convert into information, a computer is used.

BASIC COMPUTER CONCEPTS

Functions of Computers

BASIC COMPUTER CONCEPTS

Characteristics of Computer System



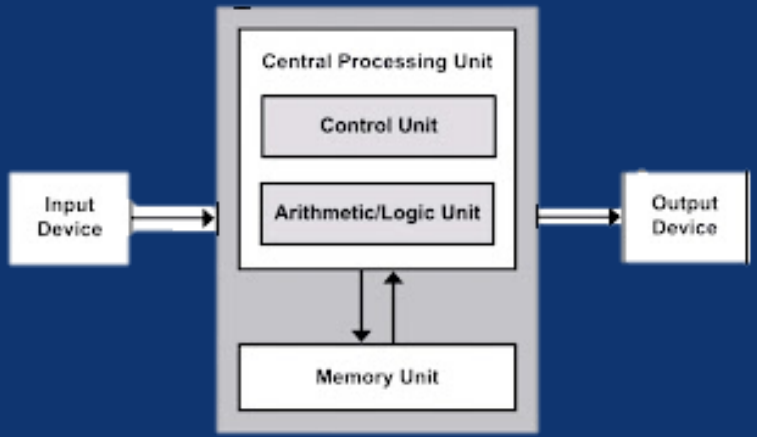
BASIC COMPUTER CONCEPTS

Basic Application of Computer

Home
Medical Field
Entertainment
Industry
Education
Government
Banking
Business
Training
Arts
Science & Engineering

BASIC COMPUTER CONCEPTS

Components of Computer System



BASIC COMPUTER CONCEPTS

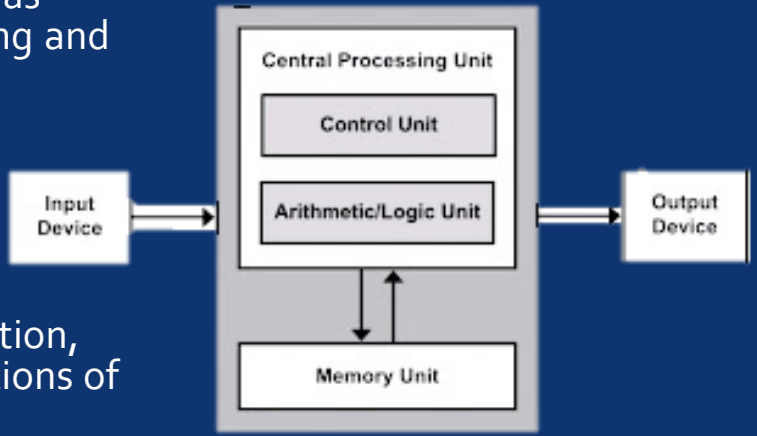
Components of Computer System

ALU

All types of processing, such as comparisons, decision-making and processing of non-numeric information takes place here

Control Unit

This part of CPU extracts instructions, performs execution, maintains and directs operations of entire system.



Functions of Control Unit

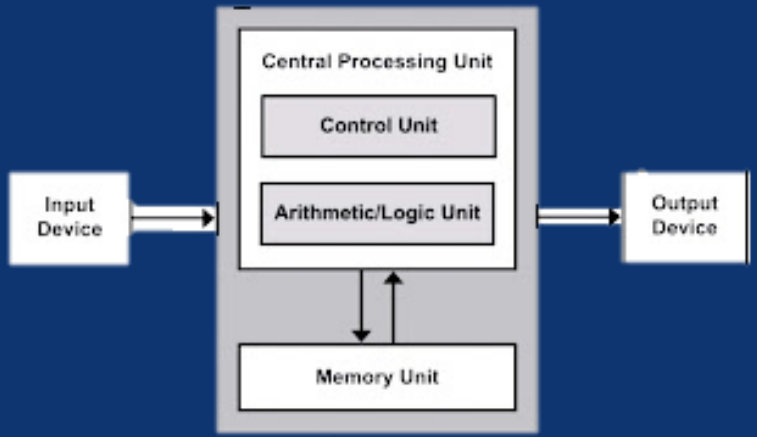
- It controls all activities of computer
- Supervises flow of data within CPU
- Directs flow of data within CPU
- Transfers data to Arithmetic and Logic Unit
- Transfers results to memory
- Fetches results from memory to output devices

BASIC COMPUTER CONCEPTS

Components of Computer System

Memory Unit

This is unit in which data and instructions given to computer as well as results are stored.



COMPUTER LANGUAGES

An overview of Computer Languages

Computer Programming Languages

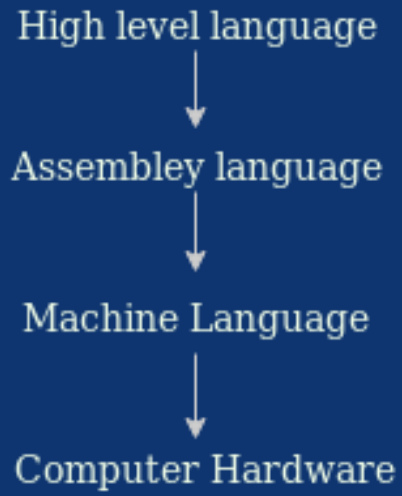
- any of various languages for expressing a set of detailed instructions for a digital computer

- allow us to give instructions to a computer in a language the computer understands

COMPUTER LANGUAGES

An overview of Computer Languages

Hierarchy of Computer language



COMPUTER LANGUAGES

An overview of Computer Languages

Language Type

- Machine and assembly languages
- Algorithmic languages (Fortran, Algol, C)
- Business Oriented Languages (Cobol, SQL)
- Object Oriented Languages (C++, C#, Ada, Java, Visual Basic, Python)
- Declarative Languages (Prolog, Lisp)
- Scripting languages (PERL)
- Document Formatting Languages (Tex, PostScript, SGML)
- WWW Display Languages (HTML, XML)
- Web Scripting (Java Script)

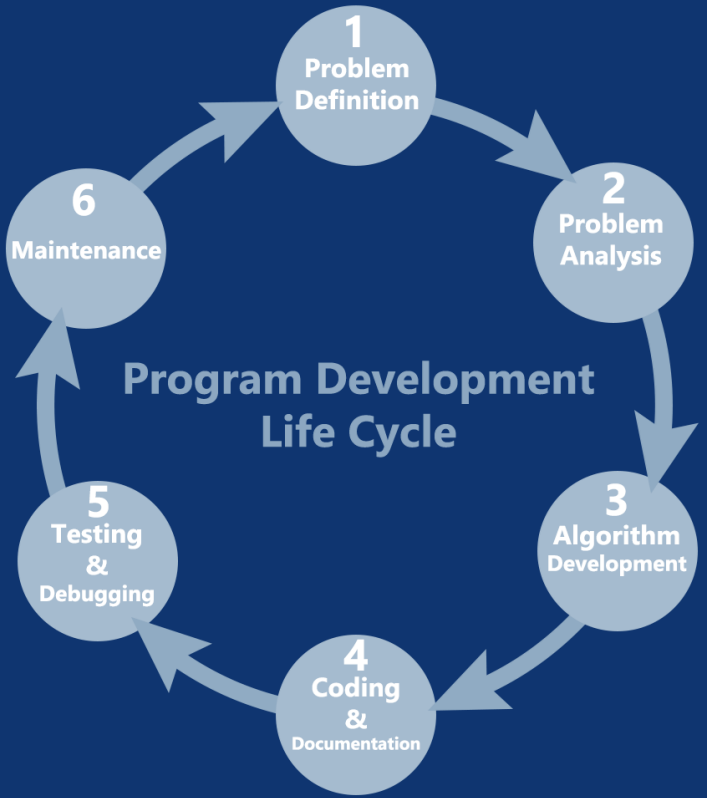
HISTORY OF C PROGRAMMING LANGUAGE

- C evolved from two previous languages, BCPL and B
- BCPL was developed in 1967 by Martin Richards
- Ken Thompson modeled many features in his B language
- In 1970 Ken Thompson used B to create early versions of the UNIX operating system at Bell Laboratories
- The C language was evolved from B by Dennis Ritchie at Bell Laboratories and was originally implemented on a DEC PDP-11 computer in 1972
- C uses many of the important concepts of BCPL and B while adding data typing and other powerful features
- C is mostly hardware independent

HISTORY OF C PROGRAMMING LANGUAGE

- By the late 1970s, C had evolved into what is now referred to as “traditional C.” The publication in 1978 of Kernighan and Ritchie’s book, The C Programming Language, drew wide attention to the language. This became one of the most successful computer science books of all time.
- The rapid expansion of C over various types of computers (sometimes called hardware platforms) led to many variations that were similar but often incompatible.
- In 1983, the X3J11 technical committee was created under the American National Standards Committee on Computers and Information Processing (X3) to “provide an unambiguous and machine independent definition of the language.” In 1989, the standard was approved; this standard was updated in 1999. The standards document is referred to as INCITS/ISO/IEC 9899-1999

PROGRAM DEVELOPMENT LIFE CYCLE



DESIGNING ALGORITHM

Algorithm

The solution to any computing problem involves executing a series of actions in a specific order. A **procedure** for solving a problem in terms of

1. the **actions** to be executed, and
2. the **order** in which these actions are to be executed

.

DESIGNING ALGORITHM

Algorithm

Example:
Give an algorithm describes your morning routine

DESIGNING ALGORITHM

What makes a good algorithm?

1. Correctness

A correct algorithm should always give the correct solution

2. Efficiency

An efficient algorithm should use a minimum resources and should preform at an acceptable speed

DESIGNING ALGORITHM

How do you measure efficiency?

Asymptotic Analysis. Refers to defining the mathematical boundation/framing of its run-time performance

DESIGNING ALGORITHM

Program Control

Specifies the order in which statements are to be executed in a computer program

DESIGNING ALGORITHM

Control Structure

Control flow is the order in which individual statements, instructions or function calls of an imperative program are executed or evaluated

Sequential:
default mode. Sequential execution of code statements (one line after another)
Selection:
used for decisions, branching. Choosing between 2 or more alternative paths.
Repetition:
used for looping, i.e. repeating a piece of code multiple times in a row.

PSEUDOCODE

- An artificial and informal language that helps you develop algorithm
- Similar to everyday English; it’s convenient and user friendly although it’s not an actual computer programming language

PSEUDOCODE

Sequential Example

START PROGRAM
ENTER TWO NUMBERS, A, B
ADD THE NUMBERS TOGETHER
PRINT SUM
END PROGRAM

PSEUDOCODE

Selection Example:

START PROGRAM
IF STUDENT'S GRADE IS GREATER THAN OR EQUAL TO 60
PRINT "PASSED"
END PROGRAM

PSEUDOCODE

Selection Example:

START PROGRAM
IF STUDENT'S GRADE IS GREATER THAN OR EQUAL TO 60
PRINT "PASSED"
ELSE
PRINT "FAILED"
END PROGRAM

PSEUDOCODE

Selection Example:

```
START PROGRAM
IF STUDENT'S GRADE IS GREATER THAN OR EQUAL TO 90
PRINT "A"
ELSE
IF STUDENT'S GRADE IS GREATER THAN OR EQUAL TO 80
PRINT "B"
ELSE
IF STUDENT'S GRADE IS GREATER THAN OR EQUAL TO 70
PRINT "C"
ELSE
IF STUDENT'S GRADE IS GREATER THAN OR EQUAL TO 60
PRINT "D"
ELSE
PRINT "F"
END PROGRAM
```

PSEUDOCODE

Repetition Example:

START PROGRAM
SET TOTAL TO ZERO
SET GRADE COUNTER TO ONE

START WHILE
WHILE GRADE COUNTER IS LESS THAN OR EQUAL TO TEN
INPUT THE NEXT GRADE
ADD THE GRADE INTO THE TOTAL
ADD ONE TO THE GRADE COUNTER
END WHILE

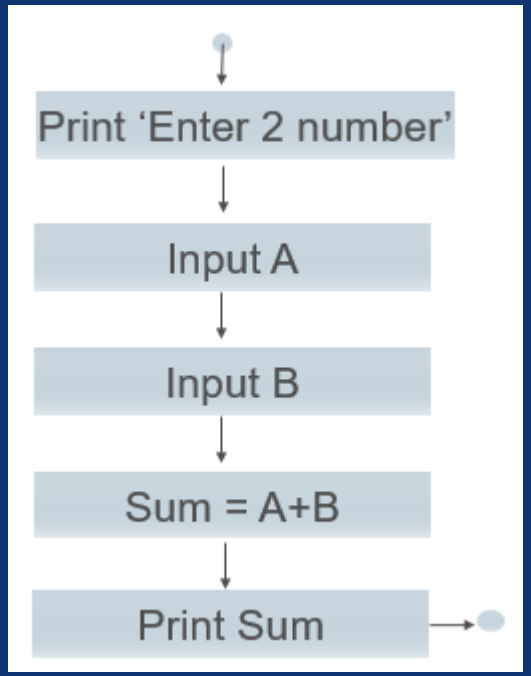
SET THE CLASS AVERAGE TO THE TOTAL DIVIDED BY TEN
PRINT THE CLASS AVERAGE
END PROGRAM

FLOWCHART

- A graphical representation of an algorithm or of a portion of an algorithm
- Drawn using certain special-purpose symbols such as rectangles, diamonds, ovals, and small circles; these symbols are connected by arrows called flowlines

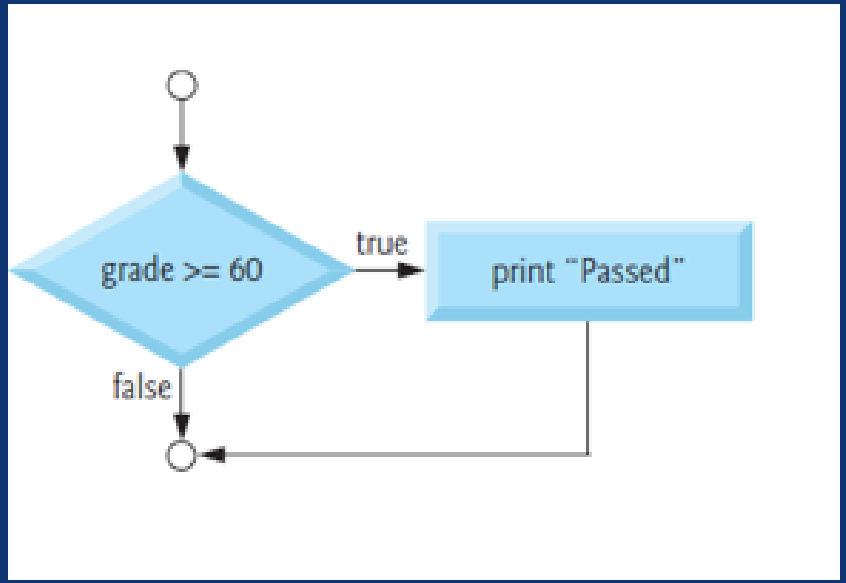
FLOWCHART

Sequential Example



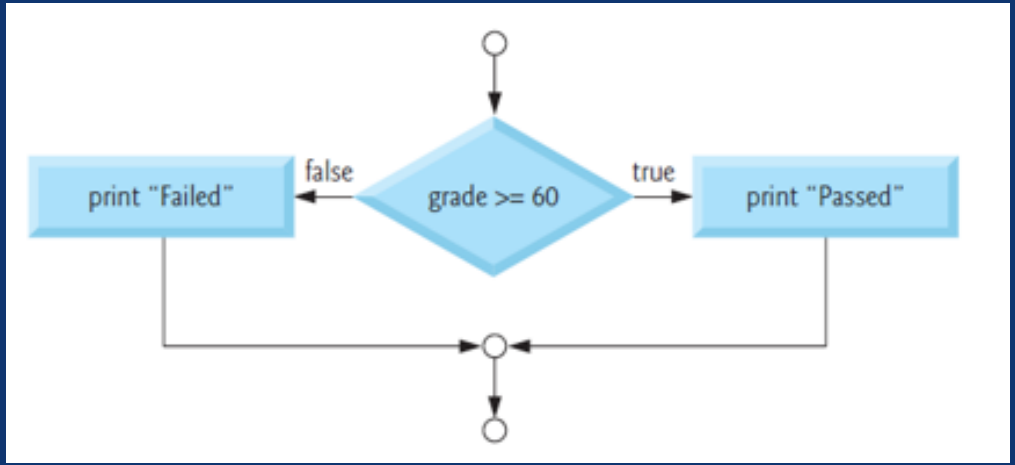
FLOWCHART

Selection Example



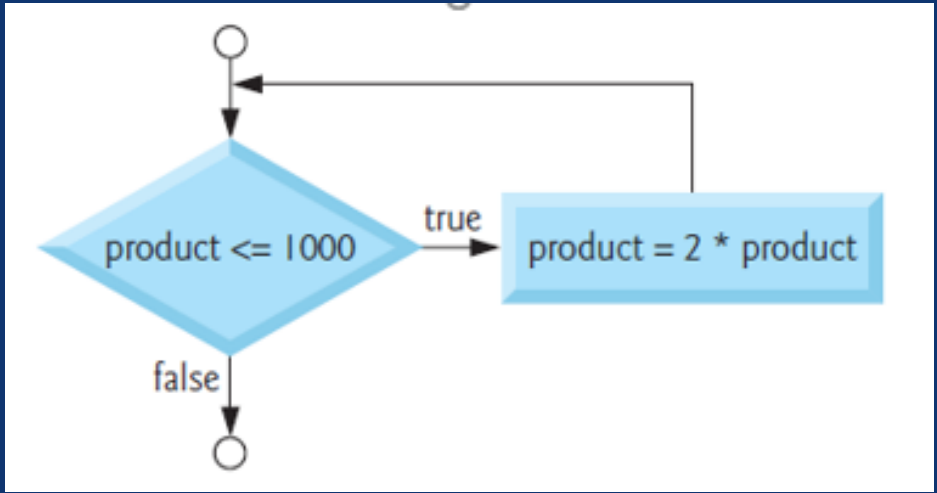
FLOWCHART

Selection Example



FLOWCHART

Repetition Example



LETURE -01

TO BE CONTINUED!

Check your LMS always for assignments and quizes

(Sometimes I forget to announce them)