



Yashveer
@yashveerSholiya

SE. ASSIGNMENT

Open & close System:-

Open System: An Open System is a System which interact with its Environment. It Receives input from the outside ~~off~~ and deliver output to outside.

close System: A close System is a System which is not interacts with Environment. Is not share information or data with its Environment.

Environment

Environment is operating ~~system~~ environment of a System -

Static & dynamic System:-

Static System: Static System are those System whose Output depends upon Only present value of Input

Dynamic System: It Means Capable of changing there Output may be variable Output depends upon present as well as past value of input.

Explain Software product & Program find Difference

PROGRAMS

PROGRAMS is a set of instructions or Commands that a Computer follow in order to perform a specific task or function, Program are the Executable Codes

Software Products:

It Contains different programs on a Computer.
It is set of instructions and its documentations to perform a specific task, Software products are of big size such as application and operating system
They are used to perform a End to End Application
Ex. Microsoft Office is a Software product. It Consist of different programs.

PROGRAM	SOFTWARE PRODUCTS
① They are usually small in size	They are larger in size
② Can be developed by single developer	Team of developer Required
③ Do not have proper USER interface	Have a User interface
④ A Program Cannot be a Software	A Software can be a program
⑤ Size Varies - (Kb) to (Mb)	Size Varies - (Mb) to (Gib)
⑥ It take less time to develop	It take more time to develop
⑦ Less no. of Codes lines in a Program	Very large Num of Codes lines in Software product
⑧ No proper Documentation	proper documentation & User Manual Provided

Discuss Early Computer programming, & also discuss the History of Computer programming

Ada Lovelace is known as First Computer programmer. In 1940 IBM has developed first electronic Computer such as IBM 602 and 604. in which fundamental programming techniques are invented and became base for programming of digital Computers

In Early 1950s,

- Computers were slow & Expensive
- Programs are very small in size that time
- It take a bit long time to Process a Program
- They Relied on assembly Language which was specified for Computer Architecture
- Developing a program required Lot of Effort.
- No Proper principles & Algorithms are used to make program Every ~~was~~ follow his own style.

Explain High Level programming Language & Low Level programming Language

High Level programmer Language :-

When Early high Level programming Language Such as "COBOL" and "FORTRAN" Came into Existence As Result

- Programming become Easier and thus increased the Productivity of Programmer
- The program ~~and~~ were limited in Size & developer Can use their own Experience or Style
- High Level language are designed to be used by human operator or programmer (User)
- High Level language programming style and Context is Easier to Learn and implement than Low level language

Features:-

- Easy to understand Similar to human language
- Program friendly Easy to Code
- Easy to Maintain
- Its portable, Easy to handle
- Can Run on any platform

Low Level language

Describe the Role of translators in High Level programming language

A Translator is a processor that Converts a Computer program from one language to another, with the help of translator we can Convert a high level language into low level language OR assembly language into machine language.

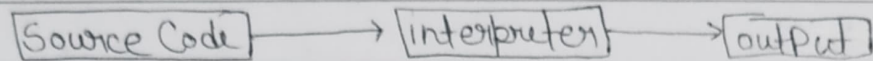
There are 3 type of translator

- (i) Interpreter
- (ii) Compiler
- (iii) Assembler

* In Interpreter & Compiler are used for high Level Language and assembler is for low level Language

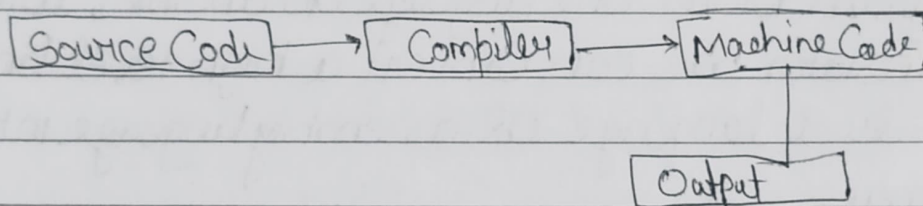
* Compiler Converts the whole high level language program to machine language at a time while interpreter Convert high level language program to machine language line by line and assembler Converts assembly language program to machine language

Interpreter: An Interpreter read and Execute each statement of Code written in high level language directly



- It takes time to execute program because it does not convert language into machine code or object code

Compiler: A Compiler first Convert the Source Code or Source program into machine Code & then Execute the program



Assembler: Assembler Convert Assembly language into Machine language

Interpreter	Compiler	Assembler
Ex. Ruby, Python, PHP	Ex C, C++	Used by Assembly lang.

OOPS Concept

Object: All Entities, involves in a design or program are known as object

- Object is a Physical Entity
- It allocate Memory Space when it Created
- User Can make more than one object in a class
- Object provide life to class
- Each & Every object have its own value
Ex. jaguar, BMW & Tesla

Class: A class is a template for Creating object

- A class is a Logical Entity
- A class does not allocate memory space when Created
- You can declare class only once
- Class generates Objects
- It doesn't have its own value

Ex Cars

Abstraction: is a method of hiding the implementation details and showing only the functionality

Abstraction in OOPS mean displaying only Essential or relevant information and hiding details which are not relevant to the User. Ex.

Inheritance :

- Process where one class acquires the property of Another
- The class which inherit the the properties of other is known as sub class or child class
- The class whose properties are inherited is known Superclass or parents class

EnCapsulation :

Machanism of wrapping the data in a Single Unit is known as Encapsulation

- It also a information hiding method

Ex. class, method, Variable of a program can bundled together

Polymorphism:

Polymorphism mean "many forms"

It Occurs when we have many classes that are related to each other by inheritance

Software characteristics:

characteristics are classified in 6 Component:-

- Maintainance - refers to the modification, extent functionality
- Efficiency - Saving time and do work correctly
- Reliability - ability of Software, storage, timing
- Functionality - refers to the performance of Software
- Portability Refers to developer can transfer Software one to another platform
- Usability - Refers to extent to which Software be used with ease

Data flow diagram	vs	Flow chart
1 Process Can operate parallel		Process execute one at time
2 Logic Aspect of Action		Physical Aspect of action
3 View of System at a low level		View of System at a high level
4 Defines function of a System		Flow chart shows how to make a System function