CORE JAVA - Srikanth Pinisetti

04 June 2021 07:19

SUBSCRIBE MY YOUTUBE CHANNEL :

https://www.youtube.com/channel/UCE5hPQZx1VvkEpG9UmzALBQ/playlists

Follow me on GITHUB for Codes :

https://github.com/therealsrikanth

What is a Program ? - Application Development(Calculator)-2+4
Set of Instruction are written in a SPECIFIC SEQUENCE or Formate computer to accomplish a given task.
Python-C-c++-Java-dotnet

Important Terms:
Bit - 0 or 1
1 Nibble :4 Bits
1 Byte - 8 bits
1 Word - 16 bits
1 Double Word - 32 bits
Multiple word - 64 bits or 128 bits

Types of SOFTWARE:
->System Software
    It is where the user is directly interacting with the Machine!
    Ex: Assembly Language
        O.S,Device Drivers(Mediator between Devices(Keyboards, mouse, printer, Pendrives etc) and O.S), Compilers, Interpreter.
->Application Software - Application Software used to perform particular operations using Application SOFTWARES.
    Ex: Python, Java, Gaming App, Editing Softwares etc

Different types of Computer Languages ?

->Low Level Languages(Assembly Languages) - COBAL and FORTRAN
    -System Understandable language.
    -Used to develop SYSTEM SOFTWARE

->High Level Languages
    Human Understanble language!
    Used to develop Application SOFTWARE!
    Ex: Python, Java,Dotnet, PHP etc
->Middle Level language
    High-Level P.L + Low Level P.Language
    Ex : C, C++ } UNIX, Java,C# etc

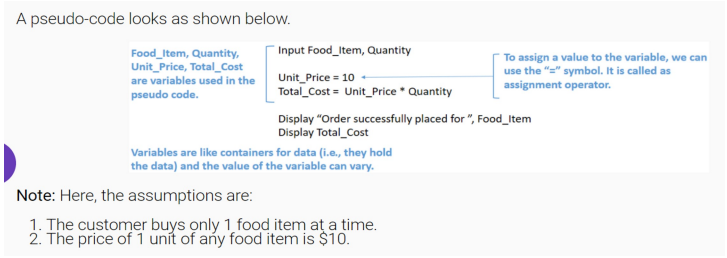
COMPILER & Interpreter

RUNTIME & COMPILE TIME ERROR (RC)

What is an Instruction or COMMAND or Function ?

```
                Instruction
                 |        |
              Command   Function - Methods
```

What is an Algorithm ?
-> Pseudo code



-> Flow Chart

Symbol	Usage	Description
→	arrowhead	represents the direction of flow
	terminal	represents the start and end of a program
	process	represents an action, process or operation
	decision	indicates a question to be answered (yes/no or true/false questions). The flowchart path can split into various branches depending on the answer
	input/output	represents the input and output of data

Control Statements ?

-> Sequential
-> Selection or Conditional(If, If-ELse, If-else-if, nested if)
-> Repeated or Iteration(loops-while, do-while, for loop)

Programming Approaches ?

-> Sequential Approach -COBAL, FORTRAN etc
-> Procedural Oriented Approach - In the form of functions(C program)
-> Object Based Approach or Object Oriented Approach - Breaking the blocks of codes - Using classes - Using Objects to run the methods in the Classes

---

What is Platform Independent and Machine Independent - JAVA ?

Platform Independent : A Program that runs on any operating System
ex: Windows, linux, MAC, UNIX
Machine Independent : Which can runs on any device
ex: Smart Devices

Java - Intro

-> Introduced by SUN MICROSYSTEMS
-> USA - 1991
-> Acquired by Oracle
-> Portable Language

C/C++ File(High-Level) -> COMPILE-COMPILER BLOCK -> Binary Code(.obj file-0or1) - O/p

Java File (name.java) -> Compiler Block -> BinaryCode (.class file)(Byte Code) -> o/p

---

JDK - Java Development Kit
JRE - Java Runtime Environment

In real time Projects or Application Development we only Compile once and execution can be done many times when ever we need.(ByteCode)

Java Editions:

SE : Standard Edition -> Develop Applications that can run only on desktops.
EE : Enterprise Edition -> Devlop Server Side Application.
ME : MicroEdition -> Develop[Application for Mobile Devices.

Naming Conventions:

-> Pascal Convention - ManojKumaran, PinisettiSrikanth - Class Names
-> camelCaseConvention - manojKumaran, pinisettiSrikanth - methods
-> SnakecaseConvention ******* - pinisetti_srikanth

JAVAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Practical Example:

CLASS - Template

[

OBJECT - Human con = new Human(); - It is a used as a link or Intermediate connection to run the logics in different methods of different classes

|
|
|

HUMAN or Dog or Car or Calculator or Restaurant (Things - Living or Non-Living)

{

DATA: State/Attributes

Weight, Height, Eye Colour, Personality, Ears etc

```
                Fuction: Behaviour
                    eat(), sleep(), work() etc
            }

            ]
```

Components of Java Code :

```
         -> Package : A container for Classes.
         -> Class : Contains Instance Variables, Methods, Local variables, Access modifiers, KeyWords etc
         -> Objects
         -> Methods or Funtions
```

JAVA Code Syntax:

```
public class FoodPlaza{

    public static void main(String[] args){
            FoodPlaza obj = new FoodPlaza();
            obj.Food();
            System.out.println(obj.Tip);
    }
    public void Food(){
        System.out.println("Pizza");
    }
    public String Tip(){
        //System.out.println("100 Dollars");
        String tip = "100 Dollars";
        return tip;
    }
}
```

Identifiers and Keywords:

```
-> Keywords:
    class
    return
    if
    if-else-if
    import
    new
    for
    while
    do
    do-while

-> Identifiers:
    vardata
    Vardata
    Var_data
    VarData
```

Methods :

```
-> Passing Parameters to a Method.
-> Returning values from a Method.
-> Local Variables.
```

Constructor :

```
-> Used to Pass the values directly.
-> Parameterless Constructor.
-> Parameterized Constructor.
```

This - Keyword :

Memory Management :

Stack	Heap

Local Variables	Instance Variables
Reference Variables	Objects
Methods	

Q4 of 6

How many objects will be eligible for garbage collection after the execution of the below code?

```
public static void main(String[] args) {

        Student student1 = new Student();
        Student student2 = new Student();
        Student student3 = new Student();
        Student student4 = student2;
        student3 = null;
        student1 = student3;
 }
```

- ○ 0
- ○ 1
- ◉ 2
- ○ 3

Consider the Account class. How many objects will be eligible for garbage collection after the execution of the below code?

```
class Account {
    double balance;

    public static void main(String args[]) {
        Account account1=null;
        Account account2=null;
        account1=new Account();
        account2=new Account();
        account2=account1;
        account1=new Account();
        account2 = account1 = null;
    }
}
```

- ◉ 3
- ○ 2
- ○ 1
- ○ 0

Access Modifiers :

public (+)

Accessible everywhere

private (-)

Accessible only inside its own class

protected(#)

Accessible inside the same package,
and to the sub-classes in different packages

default

Accessible inside the same package
– Members created without any access specifier will have this access

The visibility of members across classes and packages are shown below.

Members accessible to	public	protected	default	private
Same class	✓	✓	✓	✓
All classes in the same package	✓	✓	✓	✗
Sub-classes in different packages	✓	✓	✗	✗
All classes in different packages	✓	✗	✗	✗