Core Java

15 July 2021 16:27

CORE JAVA - Srikanth Pinisetti

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 ?

Introduction :

The first step to write a program is to create an **algorithm**. An algorithm is a process or set of rules to be followed by the computer while solving a problem. Algorithm should be represented into a form which others can easily understand. There are primarily two ways of representing an algorithm:

- **Pseudo-code** : represents the algorithm in a way that is in between a programming language and English statements

Food_Item, Quantity, Unit_Price, Total_Cost are variables used in the pseudo code.

```
Input Food_Item, Quantity

Unit_Price = 10
Total_Cost =  Unit_Price * Quantity

Display "Order successfully placed for ", Food_Item
Display Total_Cost
```

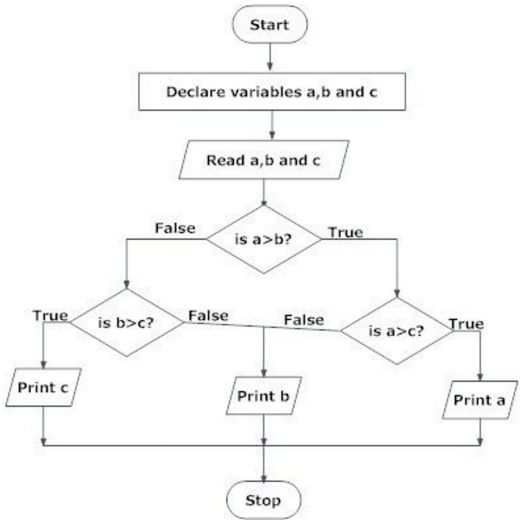
To assign a value to the variable, we can use the "=" symbol. It is called as assignment operator.

Variables are like containers for data (i.e., they hold the data) and the value of the variable can vary.

Note: Here, the assumptions are:
1. The customer buys only 1 food item at a time.
2. The price of 1 unit of any food item is $10.

- **Flowchart** : represents the algorithm in a diagrammatic way

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



❖ Problems : PSEUDOCODE

❖ Problems : FLOWCHART

Algorithm & Flowchart to find Even number between 1 to 50

Algorithm

Step-1 Start

Step-2 I = 1

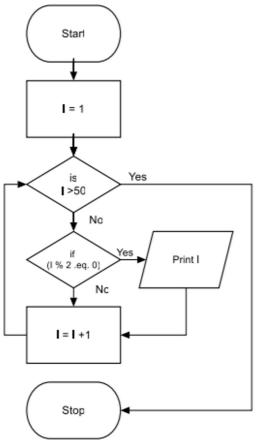
Step-3 IF (I >50) THEN
GO TO Step-7
ENDIF

Step-4 IF ((I % 2) =0) THEN
Display I
ENDIF

Step-5 I = I + 1

Step-6 GO TO Step--3

Step-7 Stop



Algorithm & Flowchart to find the largest of three numbers (an another way)

Algorithm

Step-1 Start

Step-2 Read three numbers say A,B,C

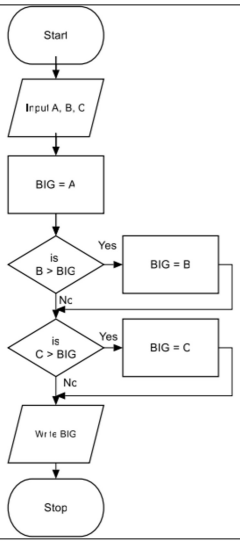
Step-3 BIG = A

Step-4 IF B > BIG THEN
BIG = B
ENDIF

Step-5 IF C >BIG THEN
BIG = C
ENDIF

Step-6 Write BIG

Step-7 Stop



Algorithm & Flowchart to find Odd numbers between 1 to n where n is a positive Integer

Algorithm

Step-1 Start

Step-2 Input Value of N

Step-3 I = 1

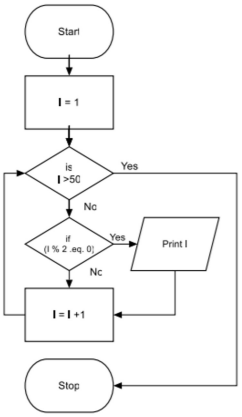
Step-4 IF (I >N) THEN
GO TO Step-8
ENDIF

Step-5 IF ((I % 2)=1) THEN
Display I
ENDIF

Step-6 I = I + 1

Step-7 GO TO Step-4

Step-8 Stop



Algorithm & Flowchart to find sum of series 1+2+3+.....+N

Algorithm

Step-1 Start

Step-2 Input Value of N

Step-3 I = 1, SUM=0

Step-4 IF (I >N) THEN
GO TO Step-8
ENDIF

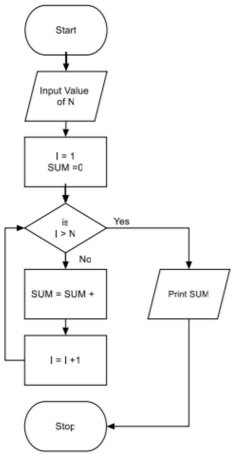
Step-5 SUM = SUM + I

Step-6 I = I + 1

Step-7 Go to step-4

Step-8 Display value of SUM

Step-9 Stop



Algorithm & Flowchart to find sum of series 1+3+5+.....+N, Where N is positive odd Integer

Algorithm

Algorithm

Step-1 Start

Step-2 Input Value of N

Step-3 I = 1, SUM=0

Step-4 IF (I >N) THEN
GO TO step 8
ENDIF

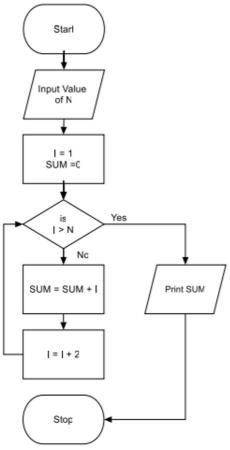
Step-5 SUM = SUM + I

Step-6 I = I + 2

Step-7 Go to step-4

Step-8 Display value of SUM

Step-9 Stop



Algorithm & Flowchart to find sum of series $1 - X + X^2 - X^3 \ldots X^N$

Algorithm

Step-1 Start

Step-2 Input Value of N, X

Step-3 I = 1, SUM=1, TERM=1

Step-4 IF (I >N) THEN
GO TO Step-9
ENDIF

Step-5 TERM = - TERM * X

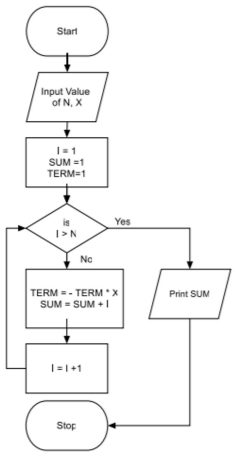
Step-6 SUM = SUM + TERM

Step-7 I = I + 1

Step-8 Go to step-4

Step-9 Display value of SUM

Step-10 Stop



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

JAVAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

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	✓	✗	✗	✗