5th Semester JAVA

Unit - I (Overview of Java)

History of Java

- Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at that time

- Java starts with the Green team, the team initiated this project to develop a language for digital device such as set-top boxes, telewision etc.

- Java was developed by James Gosling who is known as father of Java. James Gosling and his team started this project in early 1990's.

- The main name of Java — OAK.

- Java was named after OAK, Java is an island in Indonesia where the first coffee was produced (called Java coffee).

- Initially developed by James Gosling at Sun Microsystem and released in 1995.

1st Version → JDK (1996)
↳ Sun microsystem

latest version → JDK 16 (2021
↳ Oracle

Definition :-

Java is a class-based, object oriented programming language that is designed to have as few implementation dependencies as possible. It is intended to let application developers Write Once and Run Anywhere (WORA).

- Java the programming language is divided into :-

(i) Core Java	(ii) Advanced Java	(iii) Android Java
↓	↓	↓
Software development	web designing	Application

Why to learn Java

- Simple and Easy
- Open Source
- Platform independent
- Secure
- Embedded
- Robust
- Large library
- Compiler and interpreter

(#) JDK

- JDK stand for Java Development kit

- JDK (Java Development kit) is a software development environment provided by Oracle Corporation for developing Java application.

- JDK contains
  - ↳ Java Runtime Environment (JRE)
  - ↳ An interpreter / loader (Java)
  - ↳ A compiler (javac)
  - ↳ An archiver (jar) and many more

- JDK has a private Java Virtual Machine (JVM) and a few resource necessary for the development of Java application.

- Most popular JDK is Oracle JDK.

(#) JVM

- JVM stands for Java Virtual Machine

- JVM (Java virtual machine) is a platform independent programming language that converts Java bytecode into machine language and executes it.

- A Java virtual machine is a virtual machine that enables a computer to run Java programs.

- JVM is the one that calls the main method present in a Java code

- JVM is a part of JRE (Java Runtime Environment)

JRE

- JRE stands for Java Runtime Environment

- JRE is a software package that provide the necessary component to run Java application

- It is one of most common environment available on device for running java programs

- By installing the JRE, we can run Java applications, but if we want to develop Java program we need JDK (Java development kit)

* Java Syntax

```
Class _ Class name
  {
    Public static Void main (string [] args);
    {
       // code ;
    }
  }
```

Class - Predefined
class name - user define

Ex :-
```
Class _ Hello World
  {
    Public static void main ( String [ ] args ) ;
    {
      System. out. print (" Hello World") ;
    }
  }
```

Output - Hello World

* Implementing a Java program

(i) Creating a program	-	Notepad
(ii) Compiling a program	-	. class or . java c
(iii) Running a program	-	Java - interpreter

Java vs C++

Java

- Java was developed by James Gosling at sun Microsystem
- first seliase in 1995
- Platform - independent
- Java is both complide and interpreated language
- It doesn't have any virtual KeyWord
- Java uses system.in for input and system.out for output

C++

- C++ was developed by Bjarne Stroustrup at Bell labs
- first relase in 1985
- Platform - dependent
- C++ is compiled language
- C++ uses cout for input and cin for output
- It has virtual keywords
- C++ uses cin. for input and cout. for output

Data types

- It specify the different type of value that are stored in the variable
- Data types are important in programming because they determine how data can be manipulated and also affect memory allocation.

Types

① Primitive Data type

- Primitive data type are only single values and have no special capabilities. There are 8 types of primitive data type

↳ Boolean	-	true or false
↳ byte	-	two's complement
↳ char	-	unicode character
↳ short	-	two's complement
↳ int	-	two's complement
↳ long	-	two's complement
↳ float	-	floating point IEEE
↳ double	-	floating point IEEE

② Non - primitive data type

Non - primitive data type contain a memory address of variable values. Types of non - primitive data types :-

↳ string
↳ class
↳ object
↳ Interface
↳ Array

Variable

- A variable is defined as a symbolic name that is associated with a value whose value can be changed during the execution of the program

- A variable act as a storage location identified by variable name, allowing data to be stored, retrieved and manipulated efficiently

- Variable can represent different data types. such as integer, strings, boolean etc

Types of Variable

① Local Variable

A variable defined within a block or method or constructor is called a local variable. Local variable are created at the time of declaration and destroyed after existing from block.

② Instance Variable

Instance variable are declared in a class, these variables are created when an object of class is created and destroyed when object is destroyed

③ Static Variable

These variable are declared similar to instance varible. The difference is that static variable are declared using static keyword within a class outside of any method, constructor or block.

- we can declare variable in Java as ;

```
            Int count ;
           ↙          ↘
        type           name
Define data type ↲       ↳ Name given to a variable
 a variable can hold
```

Ex :-

```
Class learn
{
    int a = 10 ;     // instance variable
    int b = 20 ;
  {
    static double b = 20.5 ;
   {
     Public Static Void main ( String args [ ] ) ;

            Boolean c = true
       {
            System . Out . print (a) ;
            System . out . print (b) ;
            System . out . print (c) ;
       }
     }
   }
}
```

Literals

Any constant value which can be assigned to the variable is called literals / constants.

Ex:-

```
// 100 is a constant / literal
int x = 100 ;
```

Keywords

- It is a reserve word whose meaning is already defined in java compiler we cannot use Keywords for our personal use
- Key word are case sensitive
- Basically Java has 50 types of Keyword.

Identifiers

- Identifier is the name given to variable, classes, Methods, packages and Interface etc
- All variable are identifiers but all identifiers are not variable
- Ex :- Public static void main (string [ ] args) ;

  ↳ main - Method name
  ↳ string - Predefined class name
  ↳ args - Variable name

Array

- Arrays are used to store multiple values in a single variable instead of declaring separate variable
- for declaring an array define variable type with Square bracket ([ ])
- we declare array in Java as ;

```
type [ ] array name ;
```

type of array element

name of array

Types

① Single - Dimensional Array

- These are the most common type of arrays, where elements are stored in a linear order
- // a single dimensional array

  ```
  int [] single Dim Array = { 1, 2, 3, 4, 5 }
  ```

② Multi - Dimensional Array

- Arrays with more than one dimension, Such as two- dimensional array
- // 2 D array

```
int [ ] [ ] multi Dim Array = {
                    { 1, 2, 3 }
                    { 4, 5, 6 }
                    { 7, 8, 9 } };
```

Operator

- Operator is a symbol that is used to perform operation according to the requirement.

- Operators are special symbol that perform operation on variables or values

Types of Operator

① Arithmetic Operator

Arithmetic Operator are used to perform simple arithmetic operation on primitive and non-primitive data types.

(+ , - , * , % , /)

② Assignment Operator

Assigment Operator (=) is used to assign a value to any variable (+= , -= , *= , /= , %=)

③ Relational Operator

Relational Operator are used to check for relations like equality , greter than and less than.

(== , != , < , > , >= , <=)

④ Logical Operator

Logical Operator are used to perform "AND" and "OR" operation similar to AND and OR gate in digital electronics.

(&& , || , !)

⑤ Unary Operator

Unary operator need only one operand. they are used to increment , decrement or negative a value

(++ , --)

⑥ Shift Operator

Shift Operator are used to shift the bits of a number left or right. (<< , >> , >>>)

⑦ Ternary Operator

Ternary Operator is shorthand version of if-else statement

(? , :)

⑧ Bitwise Operator

Bitwise Operator are used to perform the manipulation of individuals bits of number with any integer type

(& , | , ^ , ~)

Control flow

- Control flow refer to the order in which individual statement, instruction, or function calls are executed in a program

- Control flow are the building block of any program that dictate the flow of execution

Types of Control flow

① Conditional Statement

Conditional statement execute a block of code based on a condition • If • If - else • If - else - If • nested if

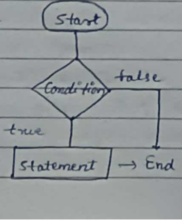
↳ If

It is used when we want to test a single condition.

Syntax -

```
if (condition)
{
   // Statement
   // Condition is true
}
```

Flow chart -



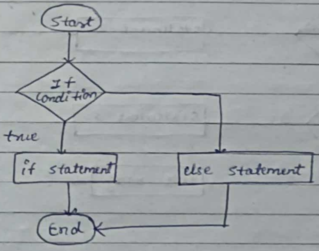
↳ If - else

It is used when we execute a program for single time

Syntax -

```
if (condition) {
        // statement
        // condition is true
    } else {
        // Statement
        // condition is false
    }
```

Flow chart -



↳ If - else - If

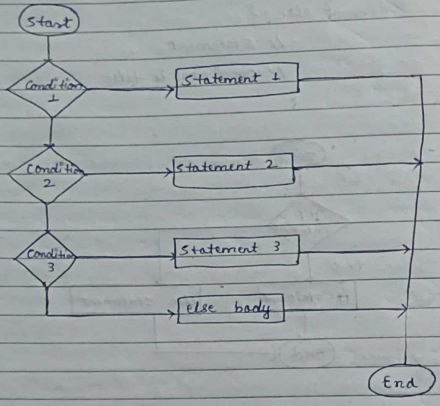
If - else - If statement allows us to execute different block of code based on multiple condition

Syntax -

```
if (condition 1) {
        // Statement 1 is true
    } else if (condition 2) {
        // Statement 2 is true
    }
```

```
else if (condition 3) {
        // Statement true
    } else {
            // If none of the condition is true
    }
```

Flow chart -



↳ Nested - if

whenever we define if block inside another if block we called as nested - if block.

Syntax -
```
if (condition 1) {
        // statement 1 is true
    if (condition 2)
        {
            // statement 2 is true
        }
}
```

② Looping Statement

Whenever we want to repeat certain statement several time then we write those statement inside loop body.

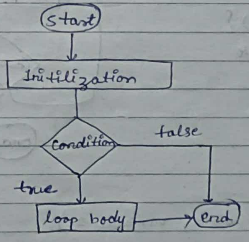
- for
- while
- do while

↳ for

when we know exactly how many times we want to loop through block of loop.

Syntax -
```
for (initilization , condition );
    {
        // code
    }
```

Flow Chart -



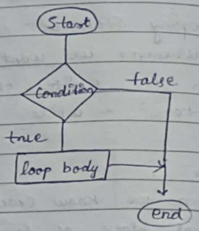
↳ while

while loop continues to execute as long as the condition evaluates to true

Syntax -

```
while (condition)
  {
   // code
  }
```

flow chart -



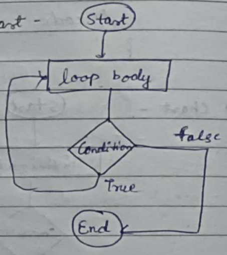
↳ do while

The code block inside a do while loop is executed at least once, even if the condition is false

Syntax -

```
do (condition 1)
  {
    // code block
  } while (condition 2) {
       // code block
       }
```

flow chart -



↳ for each

Syntax - for (data type, variable 1, variable 2)

(3) Transfer

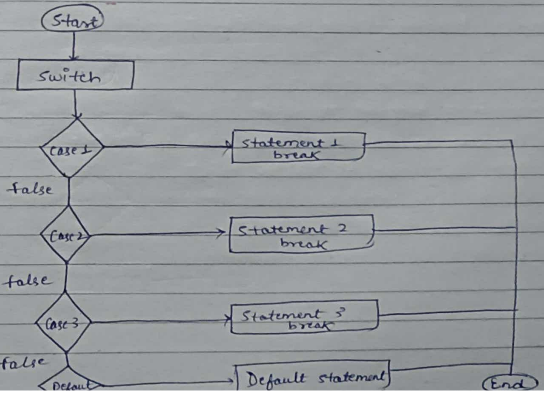
↳ Switch

switch is a multiple choice decision making selection statement it is used when we want to select only one case out of multiple case

Syntax :-

```
switch (expression)
  {
    Case 1 : Statement 1 ;
                 break;
    Case 2 : Statement 2 ;
                 break ;
    Case 3 : Statement 3 ;
                 break ;
  }
```

flowchart -


Start
Switch
Case 1
Statement 1
break
false
Case 2
Statement 2
break
false
Case 3
Statement 3
break
false
Default
Default statement
End