

HANMADP
IR

ID # 6961

“ OOP ”

—

☆

①

Q1 How many instance of variable in Java being supported by Java
Justify your answer with the help of Java coded example for each variable?

Ans TYPES OF VARIABLE:-

There are three types of variable in Java:

- Local Variable.
- Instance Variable.
- Static Variable.

1. LOCAL VARIABLE:-

A variable declared inside to body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.

2

• Instance Variable:-

↳ variable declared inside the class but outside the body of the method is called instance variable. It is not declared as static.

It is called instance variable because its value is instance specific and is not shared among instance.

• STATIC VARIABLE:-

↳ variable which is declared as static is called static variable. It cannot be local. You can create a single copy of static variable and share among all the instance of the class. Memory allocation for static variable happens only once when the class is loaded in the memory.

3

Example:-

```
Class A {  
  int data = 50; // instance variable  
  static int m = 100; // static variable  
  void method () {  
    int n = 90; // local variable  
  }  
}
```

(4)

Q2 Why 'If' is used in Java. Justify your answer with the help of Java coded example and explain in detail?

Ans:-

The Java If statement is the most simple decision-making statement. It is used to decide whether a certain statement or block of statement will be executed or not i.e. If a certain condition is true then a block of statement is executed otherwise not. Control falls into the If block.

If STATEMENT:-

If (condition)

{
// body of the If
// Statement to be executed
}

// Statement outside the If

(5)

SYNTAX:-

```
if (condition)
{
    // statement to execute if
    // condition is true
}
```

EXAMPLE:-

```
// Java program to illustrate
// if statement
```

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

```
    int i = 10;
    if (i < 15)
```

```
        System.out.println("10 is less than 15")
```

```
// This statement will be executed
// as if considers one statement
// by default
```

```
System.out.println("Outside if-block")
```

Output:-

```
10 is less than 15
Outside if-block
```

Q3:- Why "If else If" is used in Java Justify your answer with the help java coded example and explain in detail?

Ans:-

"If Else If" :-

If else If

is used to decide among multiple options. The If statement are executed from the top down. As soon as one of the conditions controlling the If is true the statement associated with that If is executed and the rest of the ladder is bypassed. If none of the conditions is true then the final else statement will be executed.

SYNTAX:-

```
If (condition)
Statement 1;
else If (condition)
Statement 2;
:
else
Statement;
```


(7)

Example:-

// Java program to illustrate
if-else-if ladder

```
import java.io.*;
```

```
class GFG {
```

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

```
{
```

```
    // initializing expression
```

```
    int i = 20;
```

```
    // condition 1
```

```
    if (i == 10)
```

```
        System.out.println("i is 10\n");
```

```
    // condition 2
```

```
    else if (i == 15)
```

```
        System.out.println("i is 15\n");
```

```
    // condition 3
```

```
    else if (i == 20)
```

```
        System.out.println("i is 20\n");
```

```
    else
```

```
        System.out.println("i is not present\n");
```

```
    System.out.println("outside if-else-if");
```

```
}
```

Output

i is 20

outside if-else-if

8

Q4 What are loops, why they are used in Java and how many types of loops are being supported by Java explain in detail?

Ans:- LOOP:-

A loop is a programming structure that repeats a sequence of instruction until a specific condition is met. Programmers use loop to cycle through values, add sums of numbers, repeat functions, and many other things.

Looping in programming language is a feature which facilitates the execution of a set of instruction/functions repeatedly while some condition evaluates to true.

Java provides three ways for executing the loops. While all the ways provide similar basic functionality, they differ in their syntax and condition checking time.

(9)

TYPES OF LOOP:-

For Loop:-

The Java for loop is a control flow statement that iterates a part of the programs multiple times.

SYNTAX:-

```
for (init; condition; inc/dec) {  
    // code to be executed  
}
```

While Loop:-

The Java while loop is a control flow statement that executes a part of the programs repeatedly on the basis of given boolean condition.

SYNTAX:-

```
while (condition) {  
    // code to be executed  
}
```


(10)

Do-while-Loop:-

The Java do while loop is a control flow statement that executes a part of the program at least once and the further execution depends upon the given boolean condition.

SYNTAX:-

```
do {  
    // code to be executed  
} while (condition);
```

Q. Write 3's table in decremented form in Java which takes input from user write Java coded program and explain in detail.

```
import java.util.Scanner;  
class Multiplication
```

```
{
```

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

```
{
```

```
    int n, c;
```

```
    System.out.println("Enter an integer  
to print its multiplication table");
```

```
    Scanner in = new Scanner(System.in);
```

```
    n = in.nextInt();
```

```
    System.out.println("Multiplication  
table of " + n);
```

```
    for (c = 3; c <= 30; c++)
```

```
        System.out.println(n + "*" + c + "  
= " + (n*c));
```

```
    }
```

8

8

(12)

Java multiplication-table. Java
Java multiplication-table

Enter number: 3

Incremented	Decrementd
$3 \times 1 = 3$	$3 \times 10 = 30$
$3 \times 2 = 6$	$3 \times 9 = 27$
$3 \times 3 = 9$	$3 \times 8 = 24$
$3 \times 4 = 12$	$3 \times 7 = 21$
$3 \times 5 = 15$	$3 \times 6 = 18$
$3 \times 6 = 18$	$3 \times 5 = 15$
$3 \times 7 = 21$	$3 \times 4 = 12$
$3 \times 8 = 24$	$3 \times 3 = 9$
$3 \times 9 = 27$	$3 \times 2 = 6$
$3 \times 10 = 30$	$3 \times 1 = 3$