

NAME: Zahoor Khan.

ID: 13835

Subject: OOP

~~Teacher~~ Teacher: Ajub Khan

I.D. 13835 Zahoor Ishaq

Q1) How many variables are being supported by Java justify your answer with the help java code example for each variable?

Java Variables

Ans: A variable is a container which holds the value while the java program is executed. A variable is assigned with a class type.

Variable is a name of memory location. There are three types of variables in Java: local, instance and static. There are two types of class types in Java: primitive and non-primitive.

"Variable"

Variable is name of reserved area allocated in memory in other words it is a name of memory location. It is a combination of "variable" that means its value can be changed.

1) `int data = 50;` Here data is variable.

Types of Variable.

There are three types of variables in Java.

(i) local variable.

(ii) Instance variable.

(iii) Static variable.

(i) local variable.

A variable declared inside the body.

of the method is called local variable. You can use this variable only within that method and

and the other methods in the class exist even where the variable exists.

A local variable cannot be defined with "static" keyword.

(2) Instance Variable

A 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 instances.

(3) Static Variable

A 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 instances of this class. Memory allocation for static variable happens only once when the class is loaded in the memory.

(Example to understand the types in Java)

(i) Class A {

(0) int data = 50; // Instance Variable

(3) static int m = 100; // Static Variable

(4) void method () {

(5) int n = 90; // local Variable

(6) System.out.println (

(7) } } output

Java Variable Example

```

(1) class Simple {
(2) public static void main (String args) {
(3) int a=10;
(4) float f=a;
(5) System.out.print (a);
(6) System.out.print (f);
(7) }

```

Output

10 10.0

Java Variable Example: Narrowing

```

(1) class Simple {
(2) public static void main (String [] args) {
(3) float f=10.5f
(4) // int a=f; // compile time error
(5) int a=(int) f
(6) System.out.print (f);
(7) System.out.print (a);
(8) }

```

Output

10.5

10

Q2 Why "if" is used in Java Justify
= Your answer with the help Java
coded example and explain in detail?

Ans: "if"

⇒ Decision making in programming is similar to decision making in real life.

In programming also we face some situation where we want a certain blocks of code to be executed when some condition is fulfilled.

Java's Selection Statement.

• if

"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.

Syntax

• if (condition)

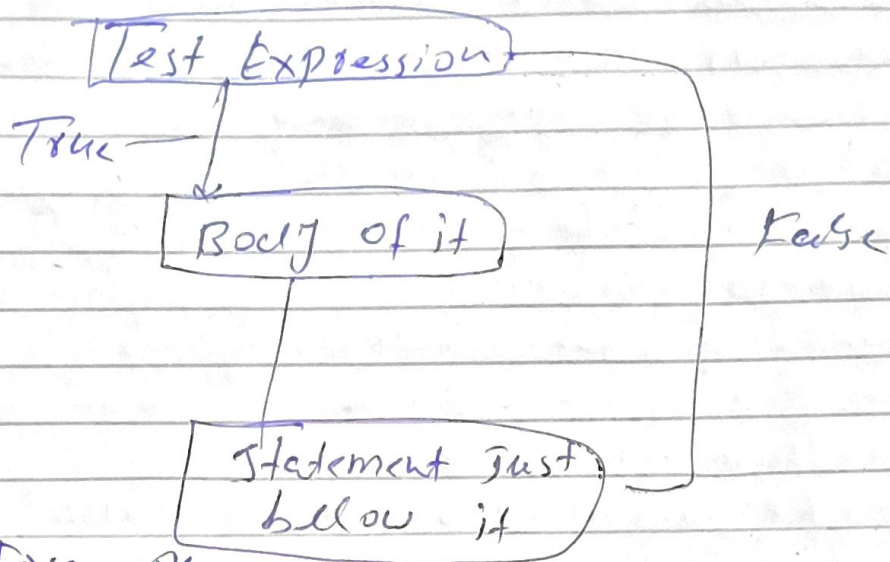
• {

• // statement to execute it

• // condition is true

• }

Flow chart-

Example

// Java program to illustrate if statement
 class: if Demo

```
{
```

```
    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 it consider one statement by default

```
            System.out.println("I am not in if");
```

```
        }
```

```
    }
```

output

I am Not in if

I.D: 13835: Zahoor Iqbal.

Q2

How if statement works.

Expression

```
int test = 5;
if (test < 10)
{
    // codes
}
// codes after if
```

Expression is false

```
int test = 5;
if (test > 10)
{
    // codes
}
// codes after if
```

Example: Java if statement.

```
class if statement {
    public static void main (String [] args) {
        int number = 10;
        // Check if number is greater is positive);
        System.out.println("This statement is
always executed");
    }
}
```

- output -

The number is positive
This statement is always executed

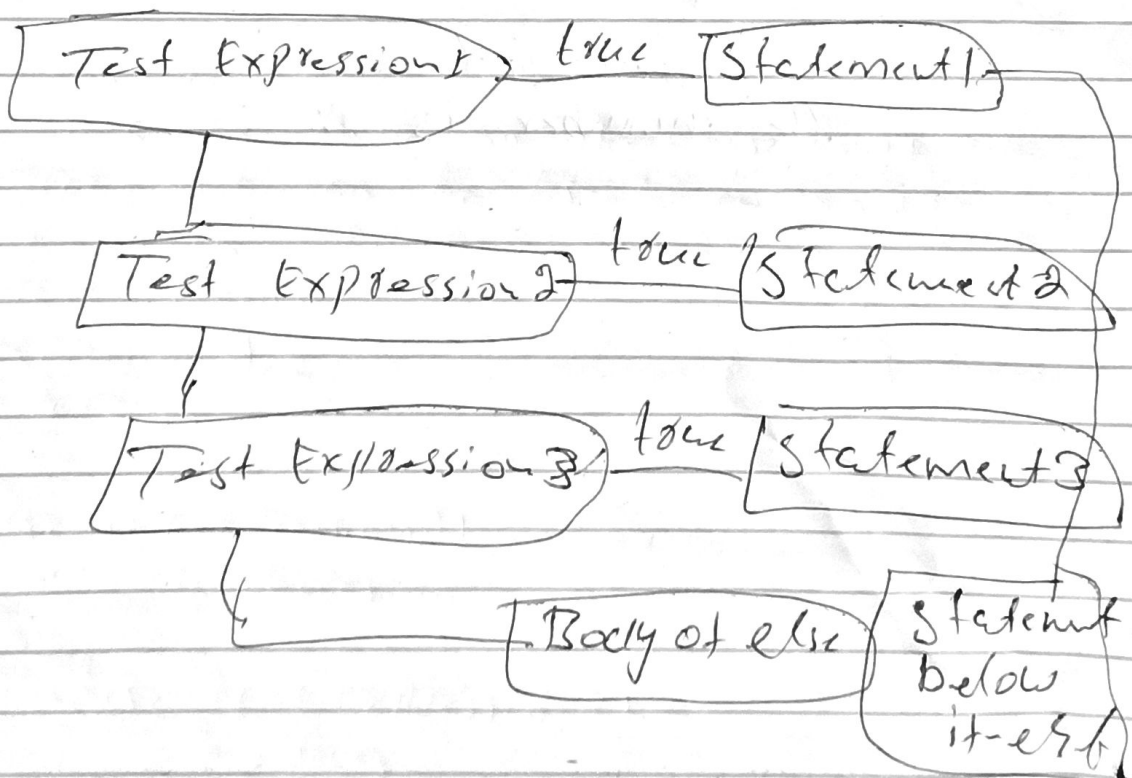
Q3 if-else-if

Here, a user can decide among multiple options. The if statements 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 code is bypassed. If none of the conditions is true, then the final else statement will be executed.

- if (condition)
- Statement
- else if (condition)
- Statement

else

Statement:



School Idhan; ID=13835

Example

```
// Java Program to illustrate it-else-if ladder
```

```
{
    public static void main (String args[])
    {
        int i = 20
        if (i == 10)
            System.out.println ("i is 10");
        else if (i == 15)
            System.out.println ("i is 15");
        else if (i == 20)
            System.out.println ("i is 20")
        else
            System.out.println ("i is not present")
    }
}
```

output:

i is 20

Java it-else-if statement.

In Java, we have an it-else-if ladder, that can be used to execute one block of code among multiple other blocks.

```
if (expression1) {
    // codes
}
else if (expression2) {
    // codes
}
else if (expression3) {
    // codes
}
```

```
else {
```

```
// codes
```

```
}
```

3

Example 3: Java if-else-if

```
class Tester {
    public static void main (String [] args) {
        int number = 0;
        // checks if number is greater than 0
        if (number > 0) {
            System.out.println ("the number is positive")
        }
        // checks if number is less than 0
        else if (number < 0) {
            System.out.println ("the number is negative")
        }
        else {
            System.out.println ("the number is 0");
        }
    }
}
```

out put

the number is 0;

T) D: 13835 Zghoor Ishaq.

Q4 loops in Java.

Loops are basically control statements. A control statement provides the way to maneuver the flow of the program into different direction that are linear otherwise. A loop is a type of control statement which encircle the flow for a while - something like the vortexes in a river stream. This article describe the concepts the behind in Java and the various types of loops.

- (1) for loop
- (2) while loop
- (3) do-while-loop

Java for loop.

A simple for loop is the same as C/C++. We can initialize the variables, check condition and increment/decrement value if consist of four parts.

(1) Initializations.

It is the initialize condition which is executed once when the loop start. Here, we can initialize the variables or we can an already initialized variable. It is an optional condition.

(2) Condition

It is the second condition which is executed each time to test the condition of the loop.

It continues execution until the condition is false. It must return boolean value either true or false. It is an optional condition.

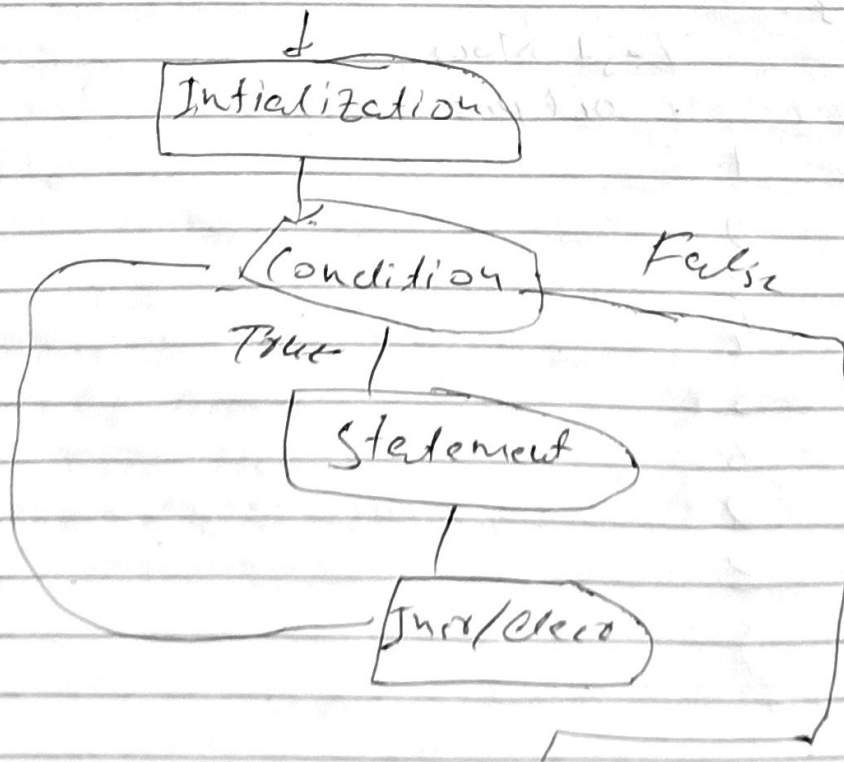
(3) Statement
The statement of the loop is executed each time until the second condition is false.

(4) Increment / Decrement
It increments or decrements the variable value. It is an optional condition.

Syntax:

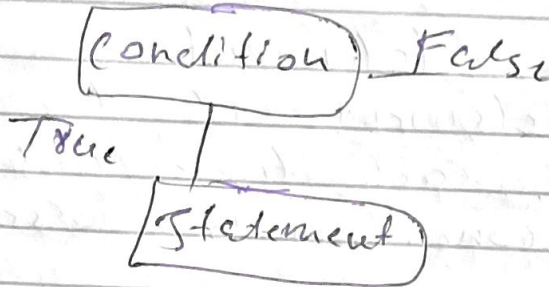
```
(1) for (Initialization; condition; incr/decr) {  
(2) // Statement or code to be executed  
(3) }
```

Diagram



Syntax

- (1) While (condition) {
- (2) // code to be executed
- (3) }



Example

```
(1) public class WhileExample {  
(2)     public static void main(String[] args) {  
(3)         int i = 1;  
(4)         while (i <= 10) {  
(5)             System.out.println(i);  
(6)             i++;  
(7)         }  
(8)     }  
(9) }
```

Test Now
out put

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

Example -

- (1) // Java Program to demonstrate the example of for loop
- (2) // which prints table of 1
- (3) public class For Example {
- (4) public static void main (String[] args) {
- (5) // code of java for loop
- (6) for (int i=1; i<=10; i++) {
- (7) System.out.println (i);
- (8) }
- (9) }
- (10) }

out put

1
2
3
4
5
6
7
8
9
10

Java while loop

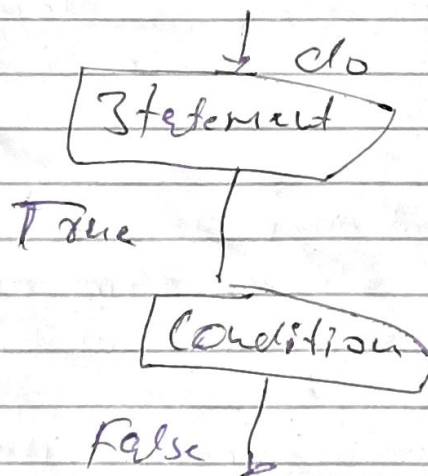
The java while loop is used to iterate a part of the program several times if the number is not fixed. It is recommended to use while loop.

Java do-while loop

The Java do-while loop is used to iterate a part of the program several times. If the number of iteration is not fixed and you must have to execute the loop at least once, it is recommended to use do-while loop.

Syntax

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



Example

```
(1) public class DoWhileExample {
(2)     public static void main(String[] args) {
(3)         int i = 1;
(4)         do {
(5)             System.out.print(i);
(6)             i++;
(7)         } while (i <= 10);
(8)     }
(9) }
```

Output	
1	7
2	8
3	9
4	9
5	10
6	

I.D: 13835. Zohoo Ishaq

Question No: 5

Ans
2

import java.util.Scanner;

public class Main {

public static void main (String [] args) {

Scanner in = new Scanner (System.in);

System.out.println ("input number:");

int n = in.nextInt();

for (int i = 10; i >= 1; i--) {

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

}

}

}

import java.util.Scanner;

public class main {

public static void main (String [] args) {

Scanner in = new Scanner (System.in);

System.out.println ("input the number:");

int n = in.nextInt();

for (int i = 10; i >= 1; i--) {

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

}

}

}

Page (16)

I.D: 13835 Zehoor Khan

Input the number 3

$$\begin{aligned} 3 \otimes 10 &= 30 \\ 3 \otimes 9 &= 27 \\ 3 \otimes 8 &= 24 \\ 3 \otimes 7 &= 21 \\ 3 \otimes 6 &= 18 \\ 3 \otimes 5 &= 15 \\ 3 \otimes 4 &= 12 \\ 3 \otimes 3 &= 9 \\ 3 \otimes 2 &= 6 \\ 3 \otimes 1 &= 3 \end{aligned}$$

Sir have already

mid marks 7 please

give marks in this
paper pass me.