JAVA FINNAL TERM ASSIGMENT

NAME:	**FAIZULLAH KHAN**
ID:	**14840**
Section:	**B**
DEPARTMENT:	**BS (SE)**
SUMITTED:	**M AYUB SIR**

DATED: 30/09/2020

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

ANS:

DEFINITION:A Java variable is a piece of memory that can contain a data value. A variable thus has a data type. Data types are covered in more detail in the text on Java data types.Variables are typically used to store information which your Java program needs to do its job.

EXPLANATION:A variable provides us with named storage that our programs can manipulate. Each variable in Java has a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable.

You must declare all variables before they can be used. Following is the

basic form of a variable declaration −

data type variable [= value][, variable [= value] ...]

Here *data type* is one of Java's datatypes and *variable* is the name of the variable.declare more than one variable of the specified type, you can use a comma-separated list.

Example:

```
int a, b, c;         // Declares three ints, a, b, and c.
int a = 10, b = 10;  // Example of initialization
byte B = 22;         // initializes a byte type variable B.
double pi = 3.14159; // declares and assigns a value of PI.
char a = 'a';        // the char variable a iis initialized
                                          with value 'a'
```

CODE VARIABLE PROGRAM EXAMPLE:

Here, *age* is a local variable. This is defined inside *pupAge()* method and its scope is limited to only this method.

```
public class Test {
  public void pupAge() {
    int age = 0;
    age = age + 7;
    System.out.println("Puppy age is : " + age);
  }

  public static void main(String args[]) {
    Test test = new Test();
    test.pupAge();
```

```
 }}
```

TYPES OF JAVA VARIABLE:

- Local variables
- Instance variables
- Class/Static variables

Local Variables

- Local variables are declared in methods, constructors, or blocks.
- Access modifiers cannot be used for local variables.
- Local variables are visible only within the declared method, constructor, or block.
- Local variables are implemented at stack level internally.

Instance Variables

- Instance variables are declared in a class, but outside a method, constructor or any block.
- Instance variables are created when an object is created with the use of the keyword 'new' and destroyed when the object is destroyed.
- Instance variables can be declared in class level before or after use.
- Access modifiers can be given for instance variables.

Class/Static Variables

- Class variables also known as static variables are declared with the static keyword in a class, but outside a method, constructor or a block.
- Static variables are stored in the static memory. It is rare to use static variables other than declared final and used as either public or private constants.
- Static variables are created when the program starts and destroyed when the program stops.

Q2. Why “If” is used in java justify your answer with the help java coded example and explain in detail?

ANS:

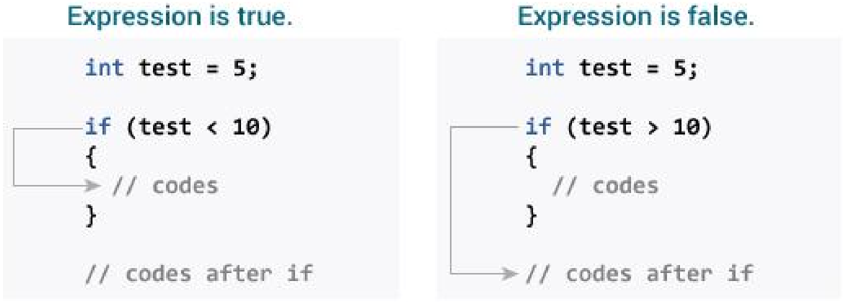
DEFFINATION:

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

EXPLANATIONS:

When we need to execute a set of statements based on a condition then we need to use control flow statements. For example, if a number is greater than zero then we want to print “Positive Number” but if it is less than zero then we want to print “Negative Number”. In this case we have two print statements in the program, but only one print statement executes at a time based on the input value. We will see how to write such type of conditions in the java program using control statements.

HOW IF STATMENT WORK IN JAVA:



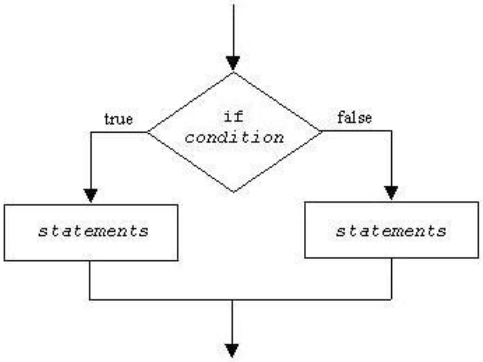
Expression is true.

```
int test = 5;

if (test < 10)
{
  // codes
}

// codes after if
```

Expression is false.

```
int test = 5;

if (test > 10)
{
  // codes
}

// codes after if
```

DIAGRAM OF (IF)STATMENT:



EXAMPLE IN JAVA CODED:

```
            public static void main(String args[]){
    int num=70;
    if( num < 100 ){
                  /* This println statement will only execute,
                  * if the above condition is true
                System.out.println("number is less than 100");
    }
        }
              }
        Output:
                  number is less than 100.
```

Q3. Why "if else if" is used in java justify your answer with the help java code example and explain in detail?

Ans:

DEFFINATIONS: if-else-if Statement:

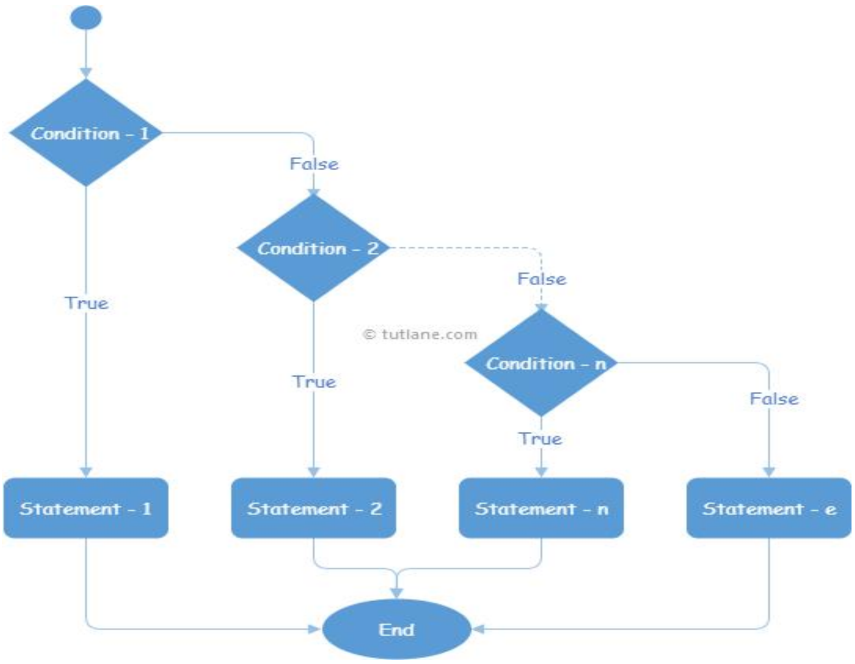
if-else-if statement is used when we need to check multiple conditions.In this statement we have only one "if" and one "else", however we can have multiple "else if". It is also known as if else if.

EXPLANATION:

The most important point to note here is that in if-else-if statement, as soon as the condition is met, the corresponding set of statements get executed, rest gets ignored. If none of the condition is met then the statements inside “else” gets executed.

When we need to execute a set of statements based on a condition then we need to use control flow statements. For example, if a number is greater than zero then we want to print “Positive Number” but if it is less than zero then we want to print “Negative Number”. In this case we have two print statements in the program, but only one print statement executes at a time based on the input value. We will see how to write such type of conditions in the java program using control statements.

DIAGRAM:



SIMPLE EXAMPLE:

```
            public class IfElseIfExample {

              public static void main(String args[]){

                   int num=1234;
                   if(num <100 && num>=1)
{
                  System.out.println("Its a two digit number");
                 }
```

CODED EXAMPLE OF IF ELSE IF:

```
                        public class IfElseIfExample {

 public static void main(String args[]){
            int num=1234;
            if(num <100 && num>=1) {
              System.out.println("Its a two digit number");
            }
            else if(num <1000 && num>=100) {
              System.out.println("Its a three digit number");
            }
            else if(num <10000 && num>=1000) {
              System.out.println("Its a four digit number");
            }
            else if(num <100000 && num>=10000) {
              System.out.println("Its a five digit number");
            }
            else {
              System.out.println("number is not between 1 & 99999");
            }
 }}
```

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: **Loops in Java:**

In programming languages, loops are used to execute a set of instructions/functions repeatedly when some conditions become true. There are three types of loops in Java.

Types of loops:

for loop
while loop
do-while loop

Java For Loop:

The Java for loop is used to iterate a part of the program several times.If the number of iteration is fixed, it is recommended to use for loop.
We can initialize the variable, check condition and increment/ decrement value.
It consists of four parts:

Initialization: It is the initial condition which is executed once when the loop starts.Here, we can initialize the variable, or we can use an already initialized variable. It is an optional condition.
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.

Statement: The statement of the loop is executed each time until the second condition is false.

Increment/Decrement: It increments or decrements the variable value. It is an optional condition.

Syntax:

```
for(initialization;condition;incr/decr){
//statement or code to be executed
}
```

Example:

```
//Java Program to demonstrate the example of for loop
//which prints table of 1
public class ForExample {
public static void main(String[] args) {
    //Code of Java for loop
    for(int i=1;i<=10;i++){
        System.out.println(i);
    }
}
}
```

JAVA WHILE LOOP:

The Java while loop is used to iterate a part of the program several times. If the number of iteration is not fixed, it is recommended to use while loop.

Syntax:

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

Example

```
 public class WhileExample {
public static void main(String[] args) {
    int i=1;
    while(i<=10){
        System.out.println(i);
    i++;
    }
}
}
```

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.

The Java do-while loop is executed at least once because condition is checked after loop body.

Syntax:

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

Example:

```
public class DoWhileExample {
public static void main(String[] args) {
   int i=1;
   do{
      System.out.println(i);
   i++;
   }while(i<=10);
}
}
```

Q5. Write 3's table in decremented form in java which takes input from user write java coded program and explain in detail?

Ans:

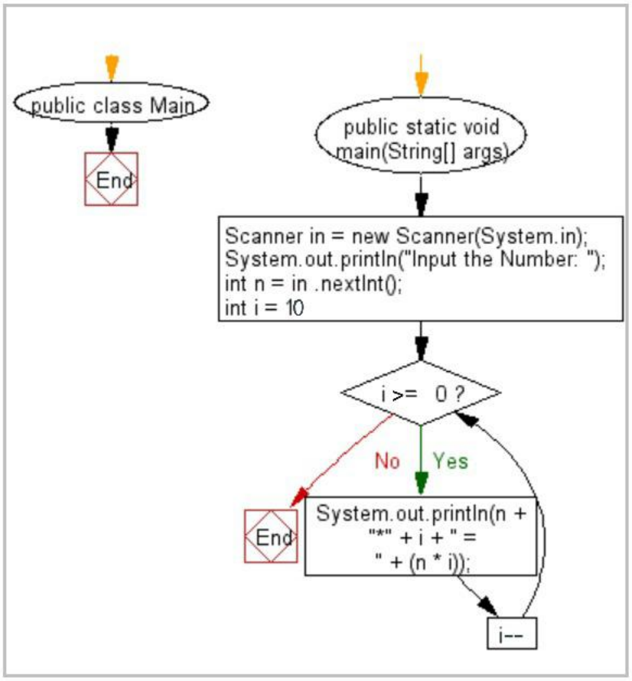
```
   import java.util.Scanner;

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();
// using for loop in decremental form
  for (int i = 10; i <= 0; I--) {
// printing in puts
   System.out.println(n + "*" + i + " = " + (n * i));
  }
 }
}
```

Output:

```
Input a number: 3
3 x 10 = 30
3 x 9 = 27
3 x 8 = 24
3 x 7 = 21
3 x 6 = 18
3 x 5 = 15
3 x 4 = 12
3 x 3 = 9
3 x 2 = 6
3 x 1 = 3
3 x 0 = 0
```

Flow chart:

Flowchart:


public class Main
End
public static void main(String[] args)
Scanner in = new Scanner(System.in);
System.out.println("Input the Number: ");
int n = in .nextInt();
int i = 10
i >= 0 ?
No
Yes
End
System.out.println(n + "*" + i + " = " + (n * i));
i--

END OF PAPER