Name: hidayatullah khan id: 14311

Subject: Object Oriented Programming

BS (CS,SE) Instructor: M.Ayub Khan

Each question carry equal marks.
Please answer briefly.

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

Three types of variables are being supported by java ,

1. Local variables
2. Instant varaibale
3. Class/static variable

Local Variables

 Local variables are declared in methods, constructors, or blocks.

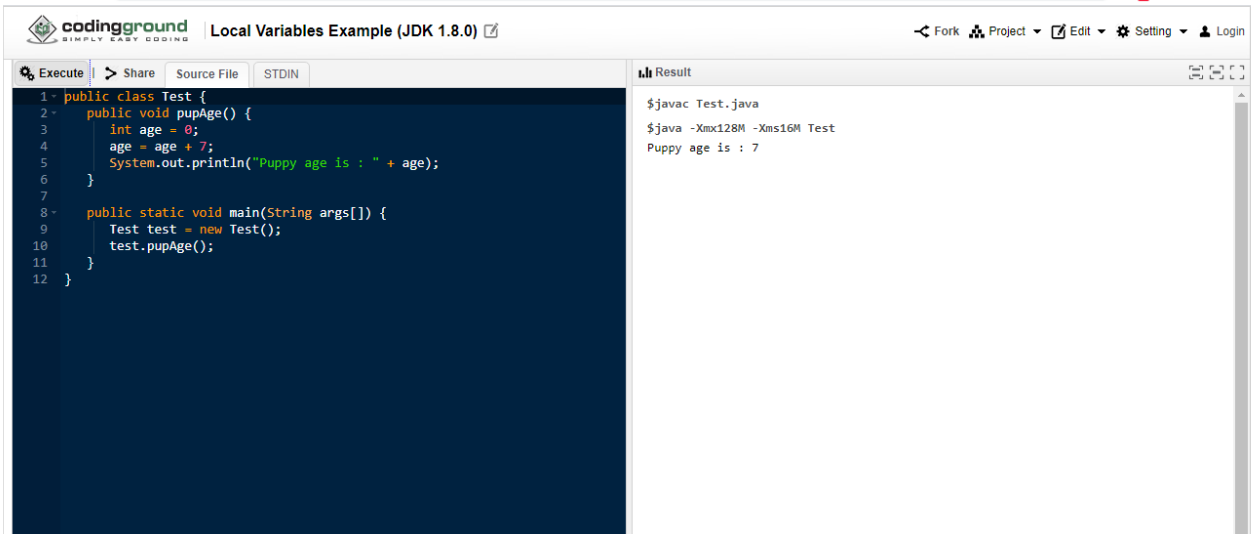
 Local variables are created when the method, constructor or block is entered and the
variable will be destroyed once it exits the method, constructor, or block.

 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.

 There is no default value for local variables, so local variables should be declared and an
initial value should be assigned before the first use.



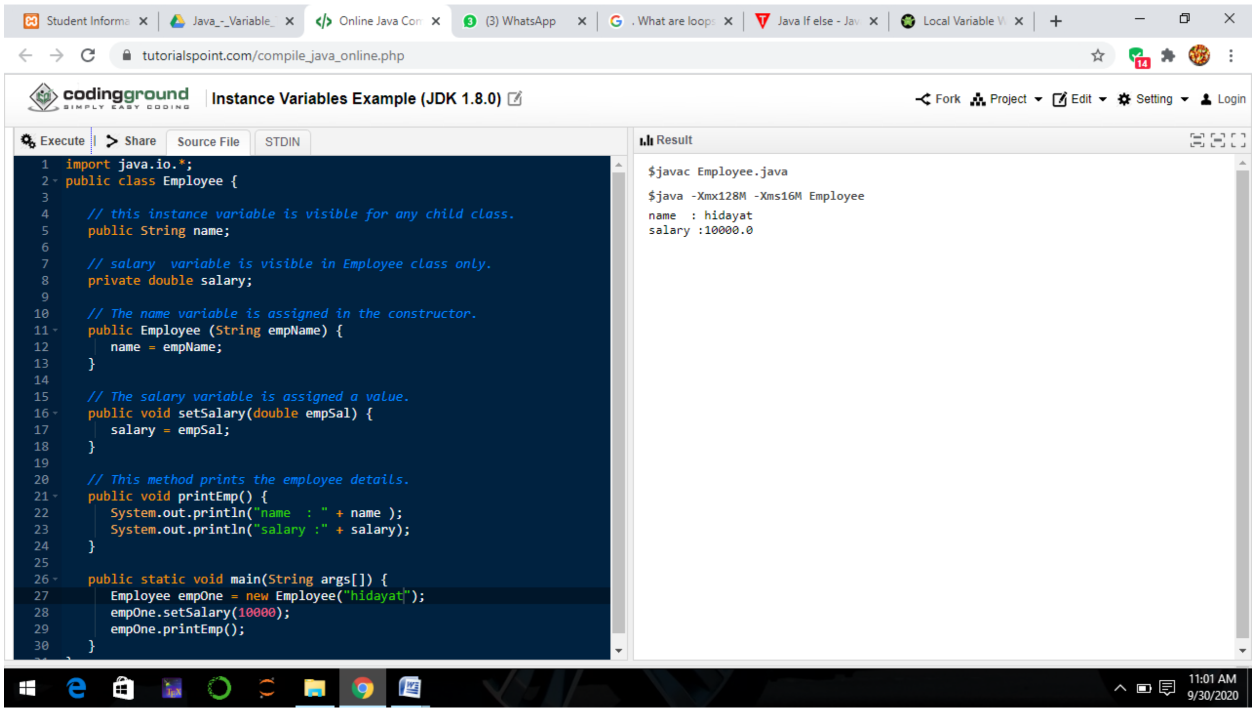
Local Variables Example (JDK 1.8.0)

```java
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();
   }
}
```

```
$javac Test.java

$java -Xmx128M -Xms16M Test
Puppy age is : 7
```

Instance Variables
 Instance variables are declared in a class, but outside a method, constructor or any block.
 When a space is allocated for an object in the heap, a slot for each instance variable value
is created.
 Instance variables are created when an object is created with the use of the keyword &#39;new&#39;
and destroyed when the object is destroyed.
 Instance variables hold values that must be referenced by more than one method,
constructor or block, or essential parts of an object&#39;s state that must be present throughout
the class.
 Instance variables can be declared in class level before or after use.
 Access modifiers can be given for instance variables.
 The instance variables are visible for all methods, constructors and block in the class.
Normally, it is recommended to make these variables private (access level). However,
visibility for subclasses can be given for these variables with the use of access modifiers.
 Instance variables have default values. For numbers, the default value is 0, for Booleans
it is false, and for object references it is null. Values can be assigned during the

declaration or within the

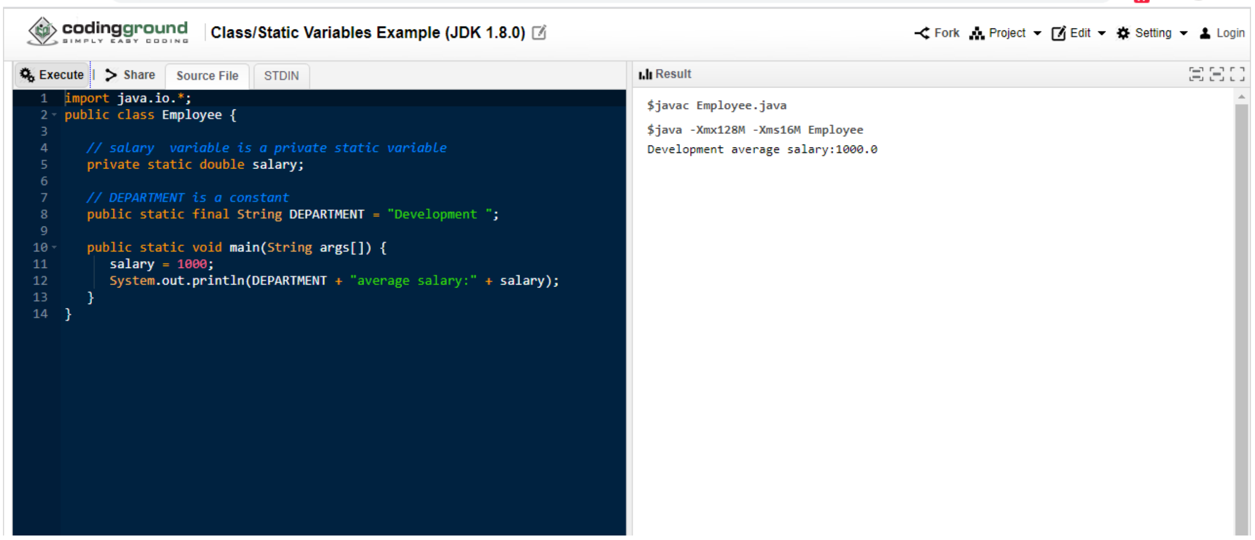

tutorialspoint.com/compile_java_online.php
codingground
Instance Variables Example (JDK 1.8.0)
Execute
Share
Source File
STDIN
Result
Fork
Project
Edit
Setting
Login

```
import java.io.*;
public class Employee {

   // this instance variable is visible for any child class.
   public String name;

   // salary  variable is visible in Employee class only.
   private double salary;

   // The name variable is assigned in the constructor.
   public Employee (String empName) {
      name = empName;
   }

   // The salary variable is assigned a value.
   public void setSalary(double empSal) {
      salary = empSal;
   }

   // This method prints the employee details.
   public void printEmp() {
      System.out.println("name  : " + name );
      System.out.println("salary :" + salary);
   }

   public static void main(String args[]) {
      Employee empOne = new Employee("hidayat");
      empOne.setSalary(10000);
      empOne.printEmp();
   }
```

```
$javac Employee.java
$java -Xmx128M -Xms16M Employee
name  : hidayat
salary :10000.0
```

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.
 There would only be one copy of each class variable per class, regardless of how many
objects are created from it.
 Static variables are rarely used other than being declared as constants. Constants are
variables that are declared as public/private, final, and static. Constant variables never
change from their initial value.
 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.
 Visibility is similar to instance variables. However, most static variables are declared

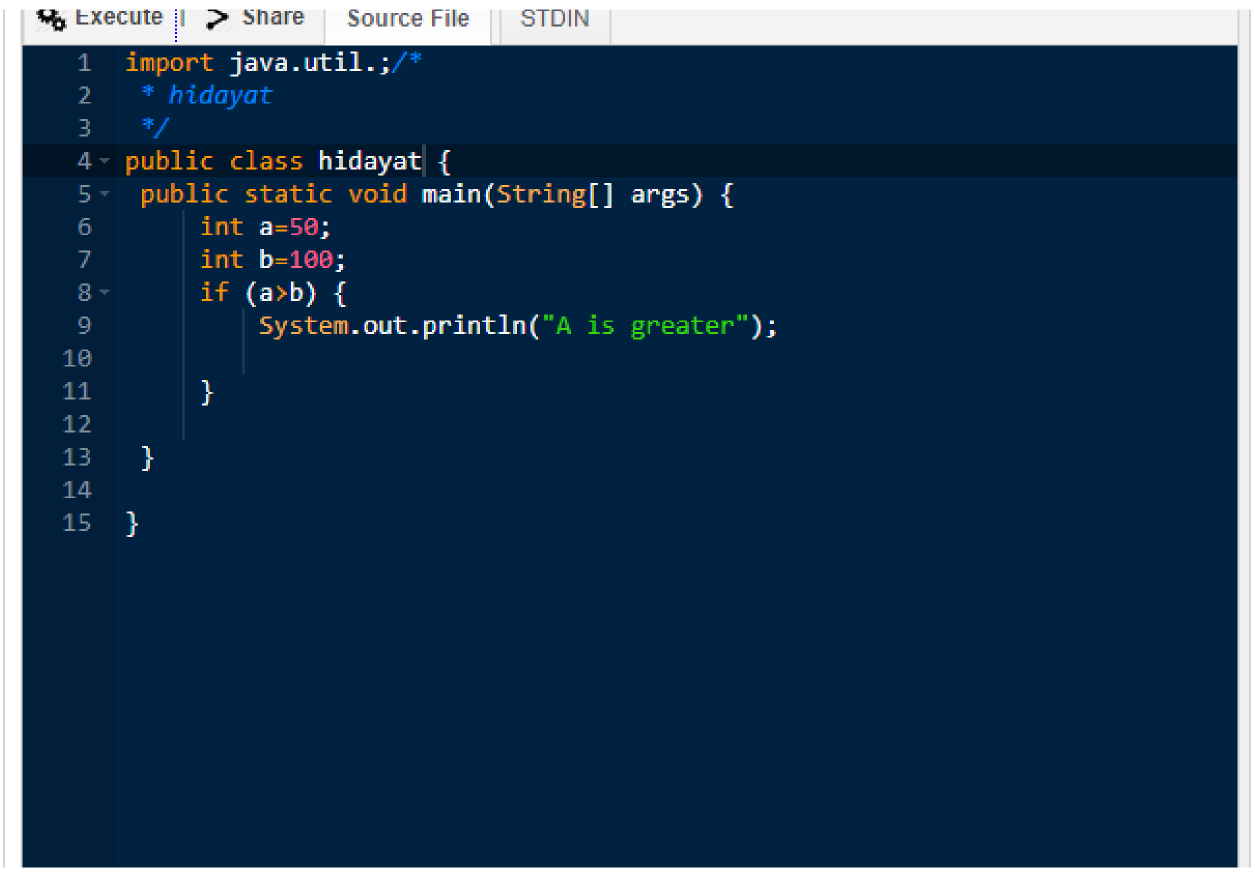
public since they must be available for users of the class.
 Default values are same as instance variables. For numbers, the default value is 0; fornBooleans, it is false; and for object references, it is null. Values can be assigned during the declaration or within the constructor. Additionally, values can be assigned in special static initializer blocks.
 Static variables can be accessed by calling with the class name ClassName.VariableName.
 When declaring class variables as public static final, then variable names (constants) are all in upper case. If the static variables are not public and final, the naming syntax is the same as instance and local variables.



```java
import java.io.*;
public class Employee {

   // salary  variable is a private static variable
   private static double salary;

   // DEPARTMENT is a constant
   public static final String DEPARTMENT = "Development ";

   public static void main(String args[]) {
      salary = 1000;
      System.out.println(DEPARTMENT + "average salary:" + salary);
   }
}
```

```
$javac Employee.java
$java -Xmx128M -Xms16M Employee
Development average salary:1000.0
```

Q2. Why “If” is used in java justify your answer with the help java coded example and explain in detail?
ANSWER:
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. Control falls into the if block.
CODE:

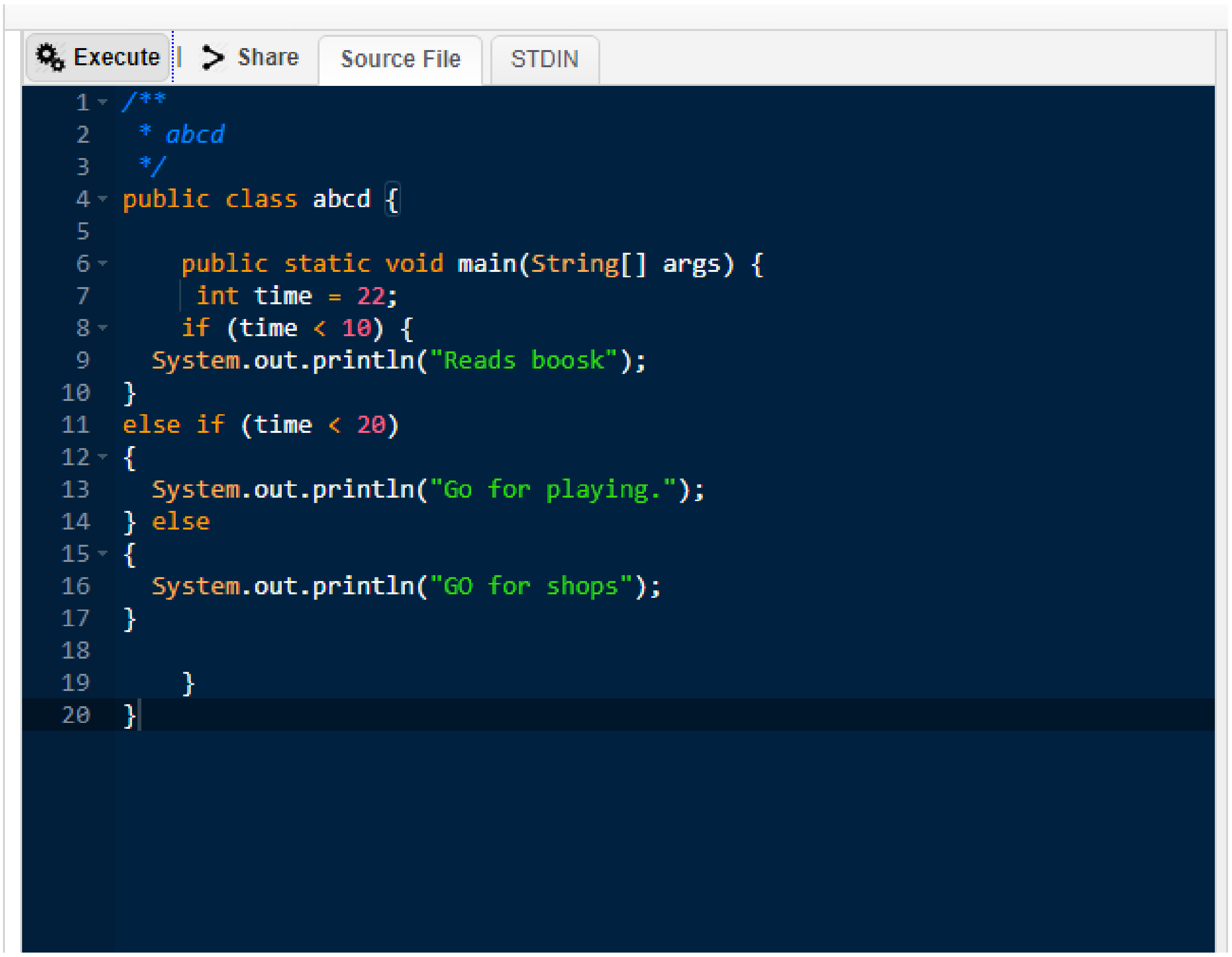


```java
import java.util.;/*
 * hidayat
 */
public class hidayat {
 public static void main(String[] args) {
     int a=50;
     int b=100;
     if (a>b) {
         System.out.println("A is greater");

     }

 }

}
```

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

ANSWER:

An if statement can be followed by an optional else statement, which executes when the boolean expression is false. 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 ladder.

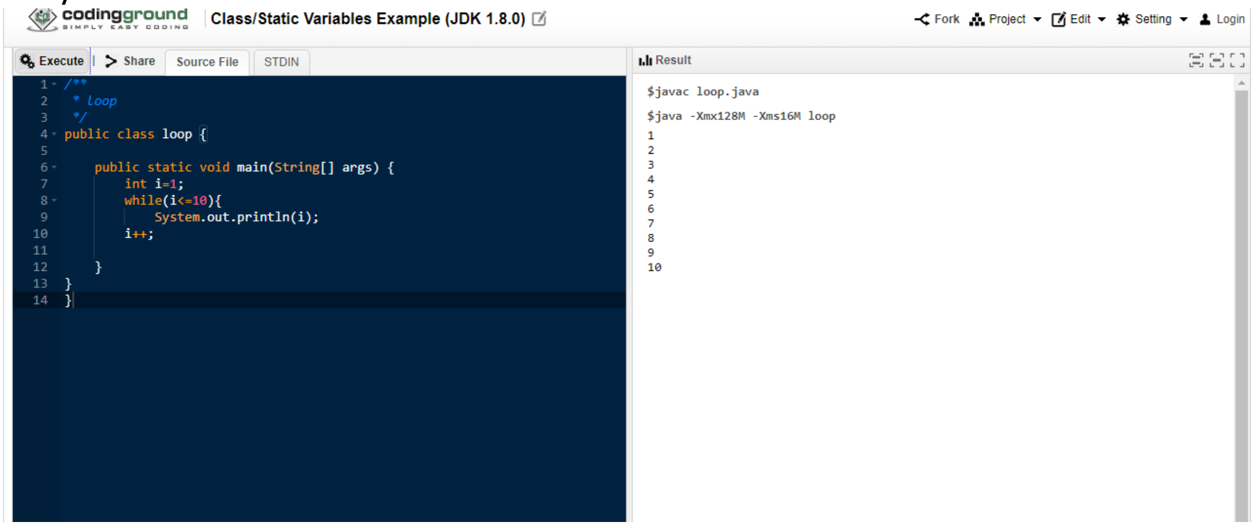


```java
/**
 * abcd
 */
public class abcd {

    public static void main(String[] args) {
     int time = 22;
    if (time < 10) {
  System.out.println("Reads boosk");
}
else if (time < 20)
{
  System.out.println("Go for playing.");
} else
{
  System.out.println("GO for shops");
}

    }
}
```

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

A loop statement allows us to execute a statement or group of statements multiple times and
following is the general form of a loop statement in most of the programming languages Java programming language provides the following types of loop to handle looping requirements. Click the following links to check their detail.
There are three types of loop;
WHILE LOOP:
Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
FOR LOOP:
Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.
DO WHILE LOOP:

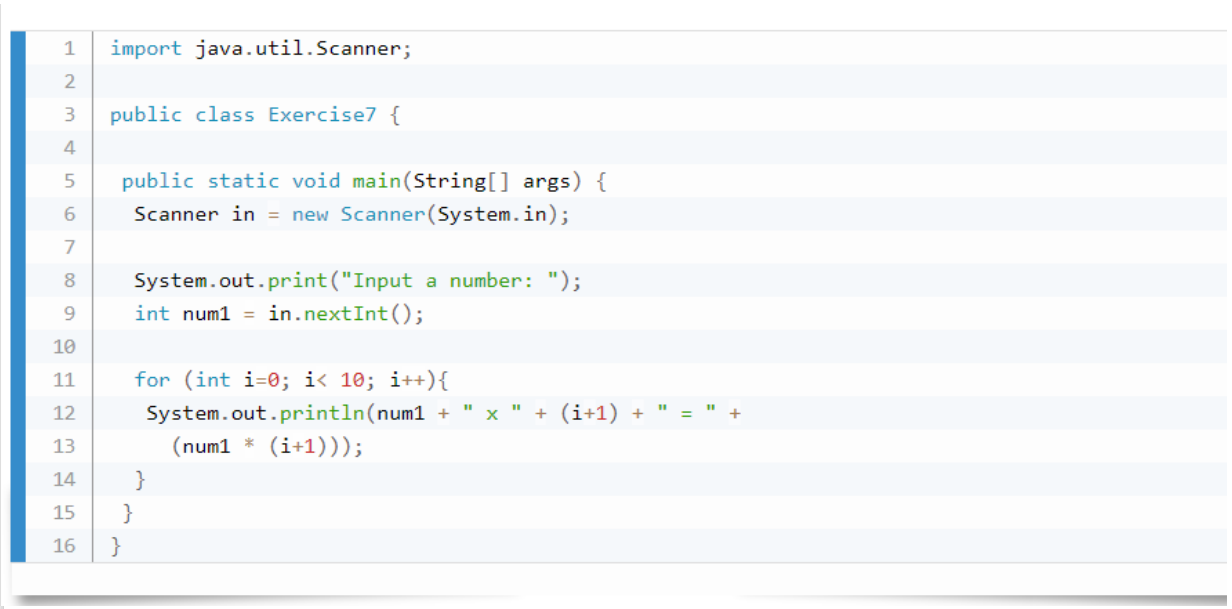
Like a while statement, except that it tests the condition at the end of the loop body.



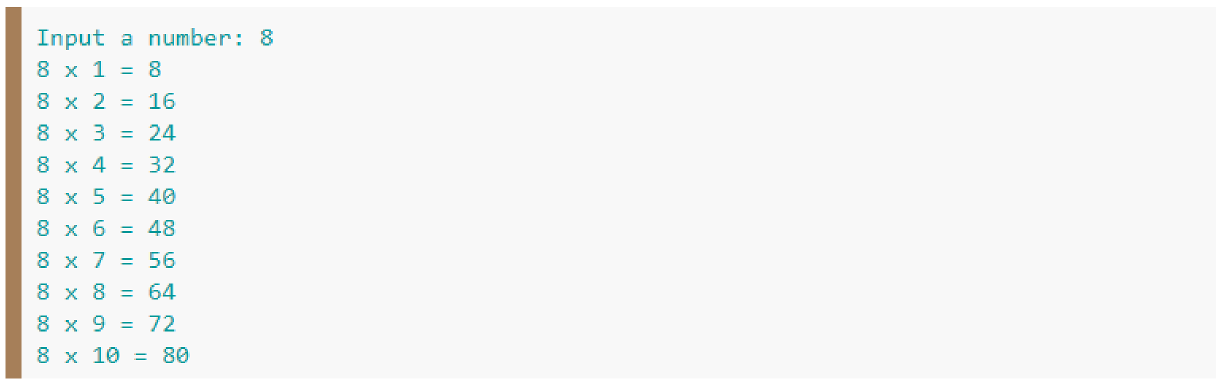
```java
/**
 * Loop
 */
public class loop {

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

    }
}
}
```

```
$javac loop.java
$java -Xmx128M -Xms16M loop
1
2
3
4
5
6
7
8
9
10
```

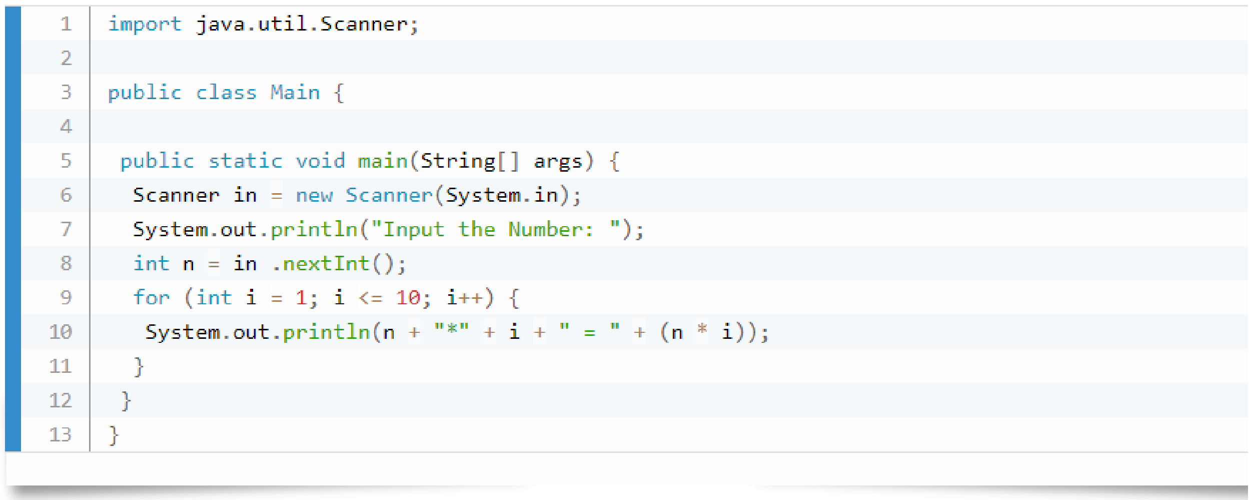
Q5. Write 3’s table in decremented form in java which takes input from user write java coded program and explain in detail?



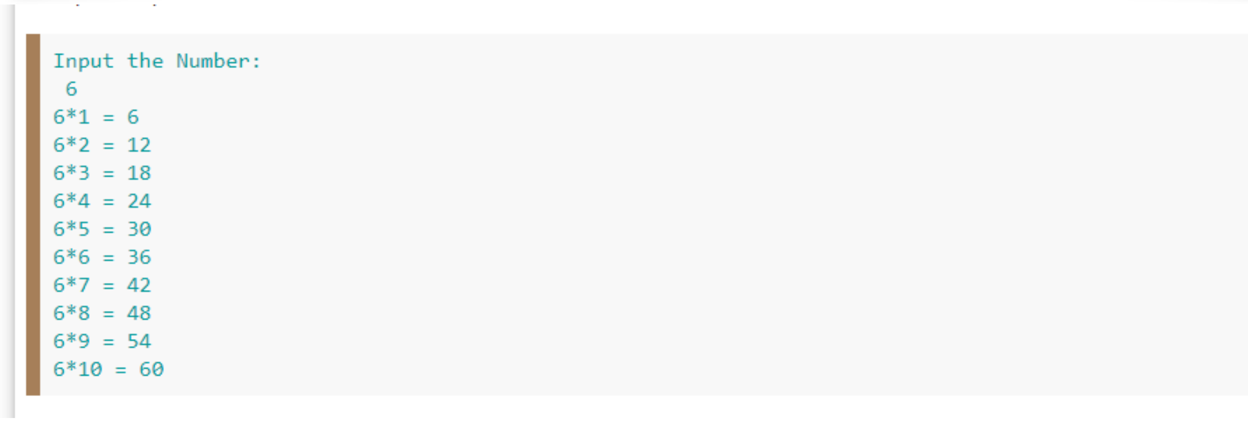
```java
import java.util.Scanner;

public class Exercise7 {

 public static void main(String[] args) {
  Scanner in = new Scanner(System.in);

  System.out.print("Input a number: ");
  int num1 = in.nextInt();

  for (int i=0; i< 10; i++){
   System.out.println(num1 + " x " + (i+1) + " = " +
     (num1 * (i+1)));
  }
 }
}
```



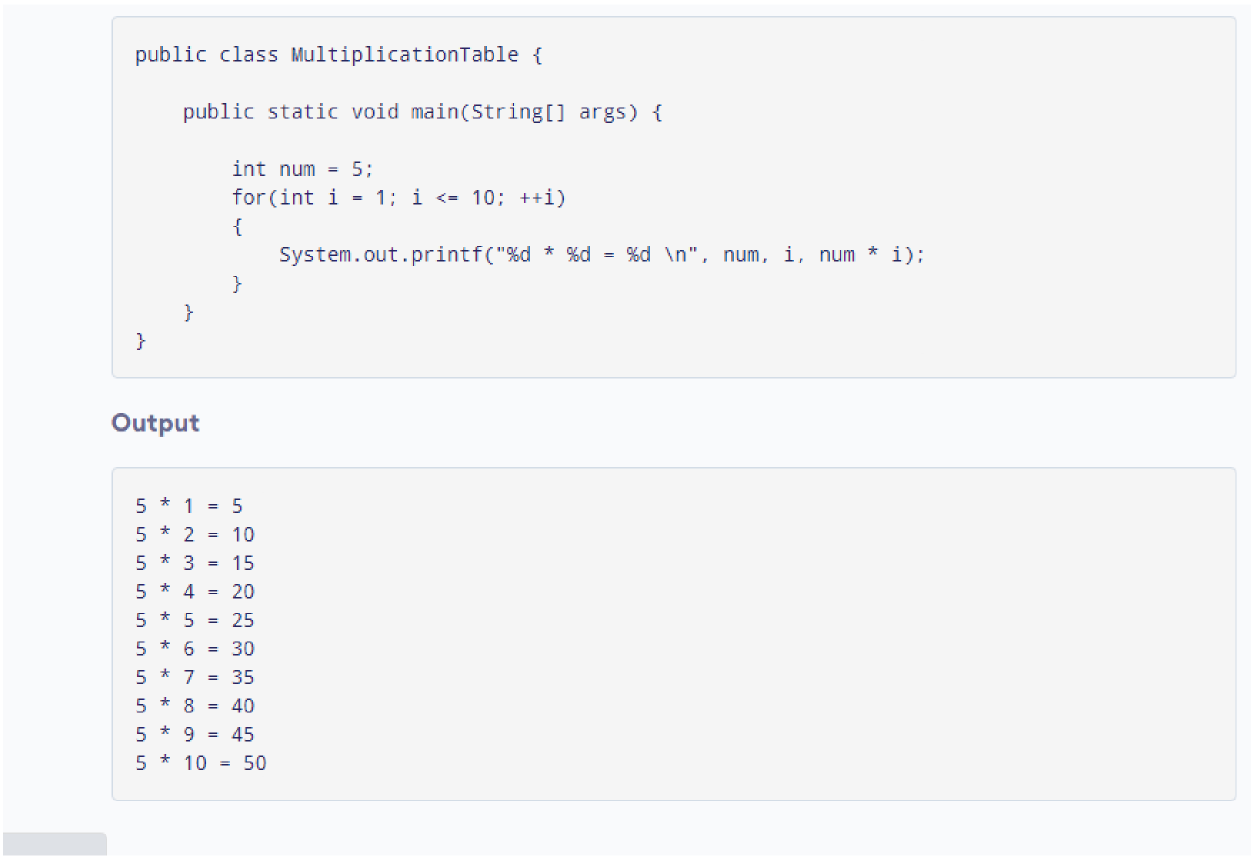
```
Input a number: 8
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
8 x 10 = 80
```



```java
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 = 1; i <= 10; i++) {
   System.out.println(n + "*" + i + " = " + (n * i));
  }
 }
}
```



```
Input the Number:
 6
6*1 = 6
6*2 = 12
6*3 = 18
6*4 = 24
6*5 = 30
6*6 = 36
6*7 = 42
6*8 = 48
6*9 = 54
6*10 = 60
```



```java
public class MultiplicationTable {

    public static void main(String[] args) {

        int num = 5;
        for(int i = 1; i <= 10; ++i)
        {
            System.out.printf("%d * %d = %d \n", num, i, num * i);
        }
    }
}
```

Output

```
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```