A method is a block of code or collection of statements or a set of code grouped together to perform a certain task or operation and they are also known as functions. It is used to achieve the reusability of code. We write a method once and use it many times.

The method, we do not require to write code again and again. It also provides the easy modification and readability of code, just by adding or removing a chunk of code. The method is executed only when we call or invoke it. You can pass data, known as parameters, into a method

Advantages of functions:

1) **Program development made easy:** Work can be divided among project members thus implementation can be completed in parallel.

2) **Program testing becomes easy:** Easy to locate and isolate a faulty function for further investigation

3) **Code sharing becomes possible**: A function may be used later by many other programs this means that a C/C++ programmer can use function written by others, instead of starting over from scratch.

4) **Code re-usability increases:** A function can be used to keep away from rewriting the same block of codes which we are going use two or more locations in a program. This is especially useful if the code involved is long or complicated.

5) **Avoid un-necessary repetition of code:** suppose you have a program that calculates square root of a number, later if you want to calculate square root of another number, you don't want to write same instructions all over again. Simply make a function that calculates square root & call it whenever need it.

6) **Increases program readability:** It makes possible top down modular programming. In this style of programming, the high level logic of the overall problem is solved first while the details of each lower level functions is addressed later. The length of the source program can be reduced by using functions at appropriate places.

7) **Function facilitates procedural abstraction:** Once a function is written, it serves as a black box. All that a programmer would have to know to invoke a function would be to know its name, and the parameters that it expects

How to create a Method?

A method must be declared within a class. It is defined with the name of the method, followed by parentheses (). Java provides some pre-defined methods, such as **System.out.println(),** but you can also create your own methods to perform certain actions:

Syntax

```
static    Return-type Name_of_Function (argument(s), ..)
          {
              Function_body ;
          }
```

❖ **static** – the keyword static means that the particular member belongs to a type itself, rather than to an instance of that type. This means we'll create only one instance of that static member that is shared across all instances of the class. A static method can be invoked without the need for creating an instance of a class.

❖ **Return Type** – A function may return a value. The **return type** is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return type is the keyword **void**.

❖ **Function Name** – this is the actual name of the function. The function name and the parameter list together constitute the function signature.
❖ **Parameters** – A *parameter* is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.
❖ **Function Body** – the function body contains a collection of statements that define what the function does.
  Note: To create a static member (block, variable, method, nested class), precede its declaration with the keyword static.

Types of functions:
A function may belong to any one of the following categories:
  1) Functions with no arguments and no return values.
  2) Functions with arguments and no return values.
  3) Functions with arguments and return values.
  4) Functions with no arguments and return values.

1) Functions with no arguments and no return value.
A java function without any arguments means you cannot pass data (values like **int**, **char** etc) to the called function and no return any type of value. Similarly, function with no return type does not pass back data to the **calling function**. It is one of the simplest types of function in java. This type of function which does not return any value cannot be used in an expression it can be used only as independent statement.
Example

```java
public class FunctionOrMethods {
    static void myMethod() {
        System.out.println("I just got executed!");
    }
    static void line(){
        for(int i=1; i<=200; i++)
            System.out.print("=");
        System.out.println("");
    }
 public static void main(String[] args) {
   myMethod();
   line();
   System.out.println("Welcome To Function ");
   line();
 }
}
```
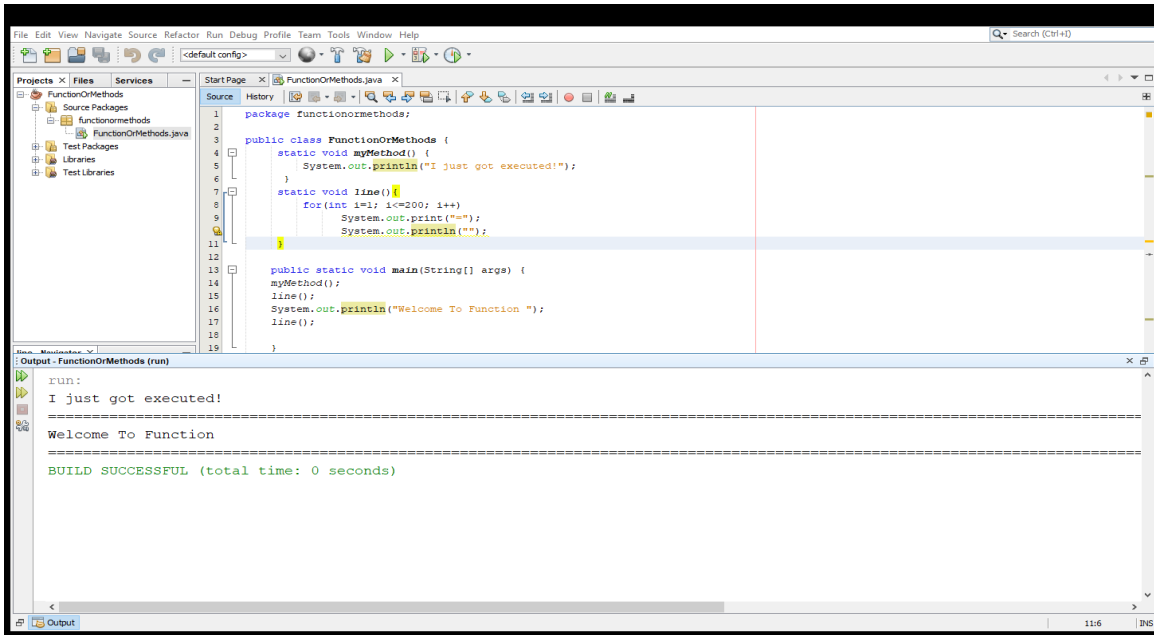
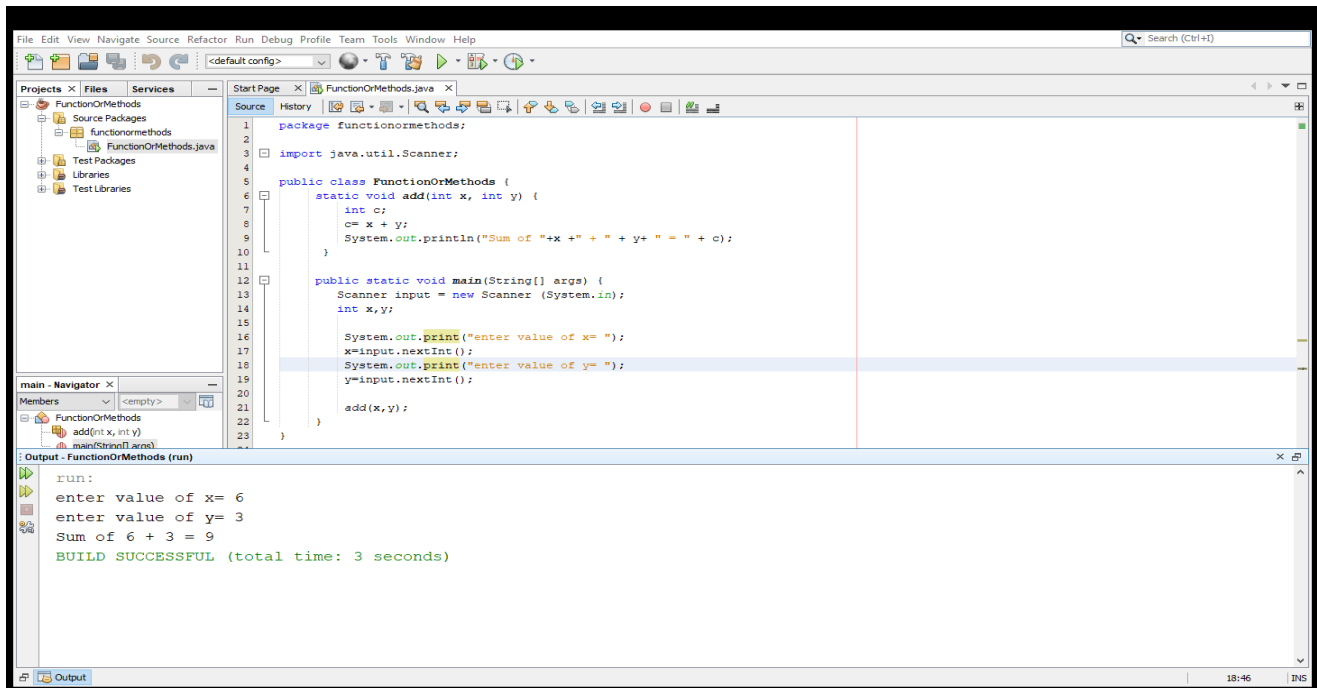## 2) Functions with arguments and no return value.

A Java function with arguments can perform much better than previous function type. This type of function can accept data from calling function. In other words, you send data to the called function from calling function but you cannot send result data back to the calling function. Rather, it displays the result on the terminal. But we can control the output of function by providing various values as arguments.

Example

```java
import java.util.Scanner;
public class FunctionOrMethods {
    static void add(int x, int y) {
        int c;
        c= x + y;
        System.out.println("Sum of "+x +" " + " " + y+ " " = " + c);
    }
 public static void main(String[] args) {
        Scanner input = new Scanner (System.in);
        int x,y;
        System.out.print("enter value of x= ");
        x=input.nextInt();
        System.out.print("enter value of y= ");
        y=input.nextInt();
            add(x,y);
    }
}
```

3. Functions with arguments and return value.

This type of function can send arguments (data) from the calling function to the called function and wait **for** the result to be returned back from the called function back to the calling function. And this type of function is mostly used in programming world because it can do two way communications; it can accept data as arguments as well as can send back data as return value. The data returned by the function can be used later in our program **for** further calculations.

Example

```java
public class ReturnFunction {
    static int add(int x, int y)          // function definition
    {    int result;
        result =    x+y;
        return(result);
    }

    public static void main(String[] args) {
        int x=30,  y=20 , z;
        z = add(x,y);                  //function call ,the return value of function will stored to z
        System.out.println("Result " +z);     // result of function will display here

    }
}
```

4. Functions with no arguments but returns value.

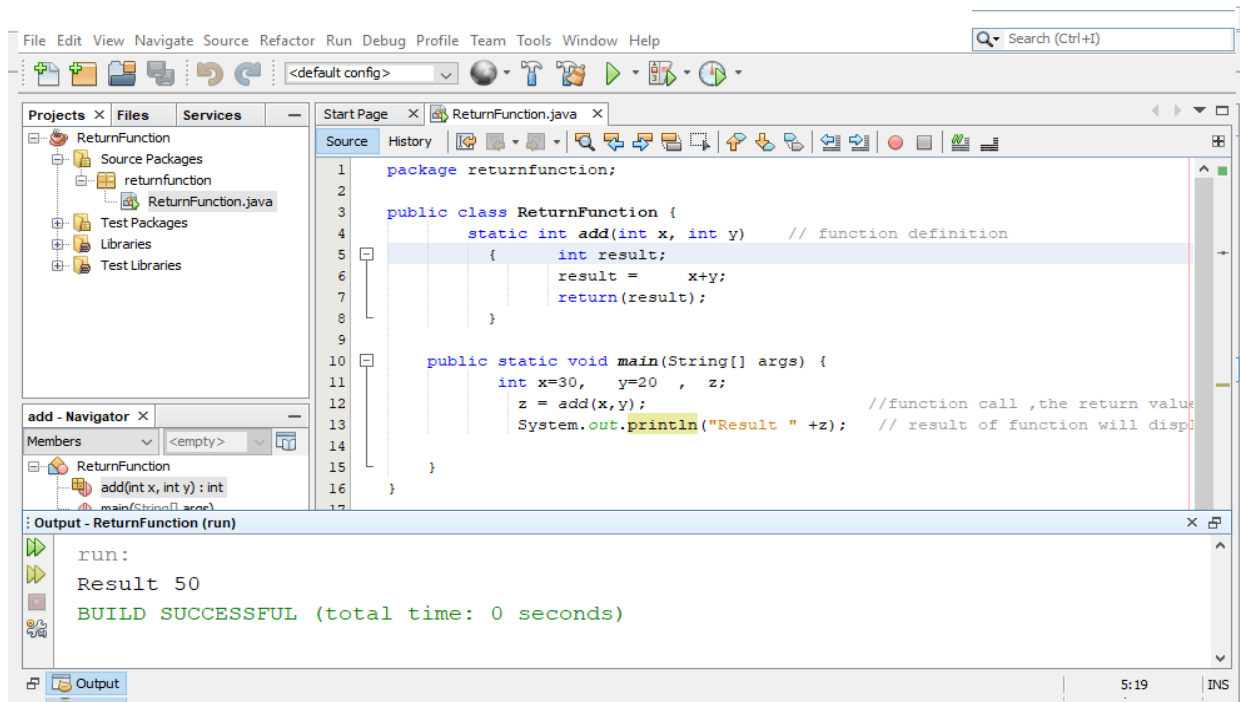This type of function which does not take any argument but only returns values to the calling function. Its mean This type of function cannot send arguments (data) from the calling function to the called function and wait only for the result to be returned back from the called function back to the calling function.

Example

```java
import java.lang.Math;
public class ReturnFunction {
    static int getRandomeNumber()     // function definition
    {                   // random number get from  1 to 100
        int b = (int)(Math.random()*(100-1+1)+1);
        return (b);
    }
    public static void main(String[] args) {
        int z;
        z = getRandomeNumber();            //function call ,the return value of function will stored to z
        System.out.println("Random Number " +z);   // result of function will display here


    }
}
```
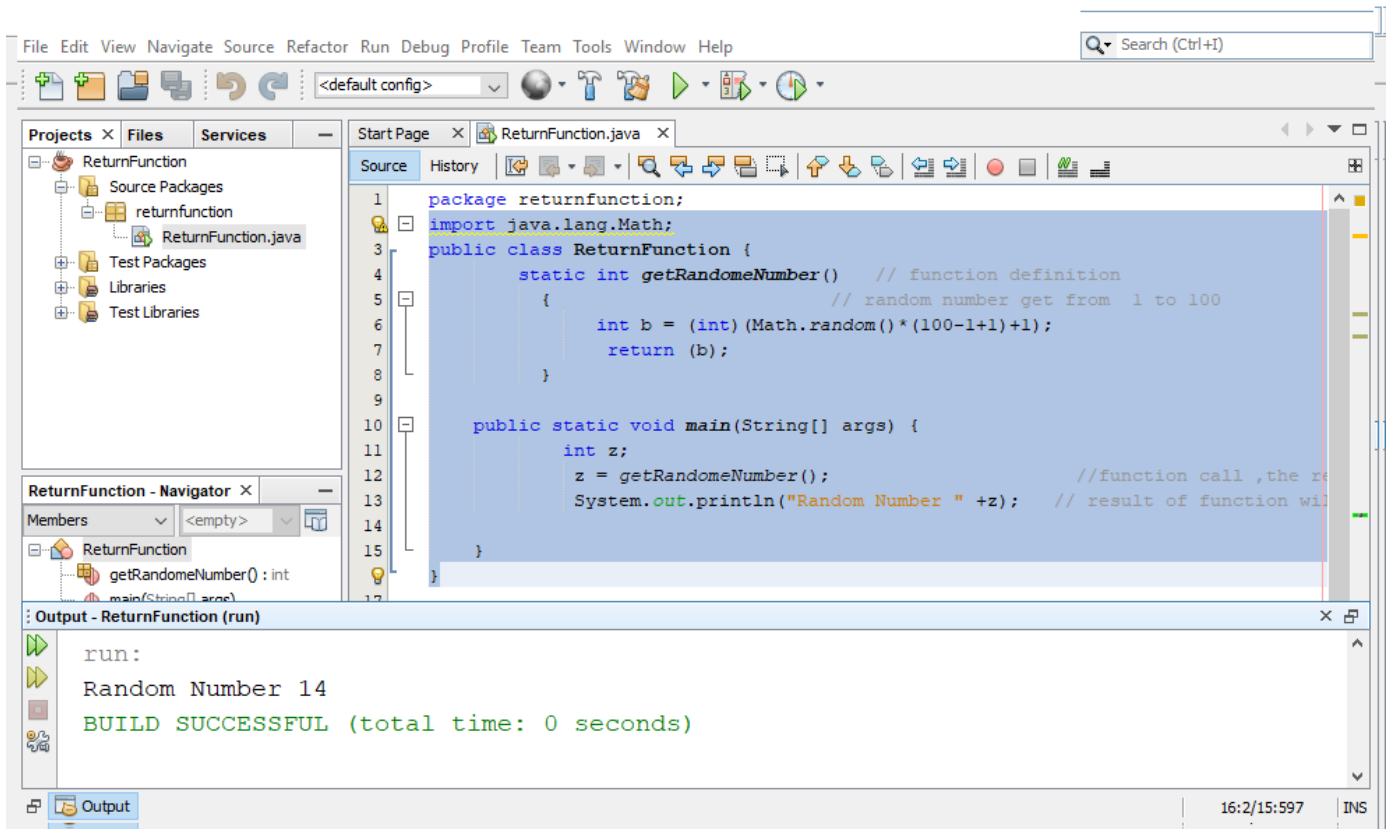
Write a program to input Student roll number, Name, Math, English and Urdu Marks and then calculate the obtain marks percentage and grade and display all information with the help of function or methods.

Example

```
import java.util.Scanner;

public class MarksheetUsingFunctions
{

    static int Roll,Math,Eng,Urdu,ObtMarks;
    static  float per;
    static  String Name,Grade;


    static void  inputData()
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter Roll Number..........: ");
        Roll = input.nextInt();
        System.out.print("Enter Student Name.........: ");
        Name = input.next();
        System.out.print("Enter Roll Maths Marks.....: ");
```

```java
        Math = input.nextInt();
        System.out.print("Enter Roll English Marks...: ");
        Eng = input.nextInt();
        System.out.print("Enter Roll Urdu Marks......: ");
        Urdu = input.nextInt();


    }
    static void processingData()
    {
        ObtMarks = Math +Eng +Urdu;
        per = ObtMarks * 100 /300;
        if (per >=80)
            Grade="A+1";
        else if(per >=70)
            Grade ="A";
        else if(per >=60)
            Grade ="B";
        else if(per >=50)
            Grade ="C";
        else if(per >=40)
            Grade ="D";
        else
            Grade ="Fail";
    }
    static void displayData()
    {   System.out.print("\n\n");
        System.out.println("Roll Number    is ........ : "+Roll);
        System.out.println("Student  Name  is ........ : "+Name);
        System.out.println("Maths Marks     is .......... : "+Math);
        System.out.println("English Marks    is .......... : "+Eng);
        System.out.println("Urdu Marks      is .......... : "+Urdu);
        System.out.println("Obtain Marks    is .......... : "+ObtMarks);
        System.out.println("Percentage is....... : "+per);
        System.out.println("Grade is  .......... : "+Grade);
    }
```

```java
public static void main(String[] args)
{
    inputData();
    processingData();
    displayData();
}
```

}

```
Output - MarksheetUsingFunctions (run)

run:
Enter Roll Number..........: 11
Enter Student Name.........: Imran
Enter Roll Maths Marks.....: 45
Enter Roll English Marks...: 56
Enter Roll Urdu Marks......: 67


Roll Number    is ........ : 11
Student  Name  is ........ : Imran
Maths Marks      is ......... : 45
English Marks    is ......... : 56
Urdu Marks       is ......... : 67
Obtain Marks     is ......... : 168
Percentage is....... : 56.0
Grade is   .......... : C
BUILD SUCCESSFUL (total time: 19 seconds)

Output
```
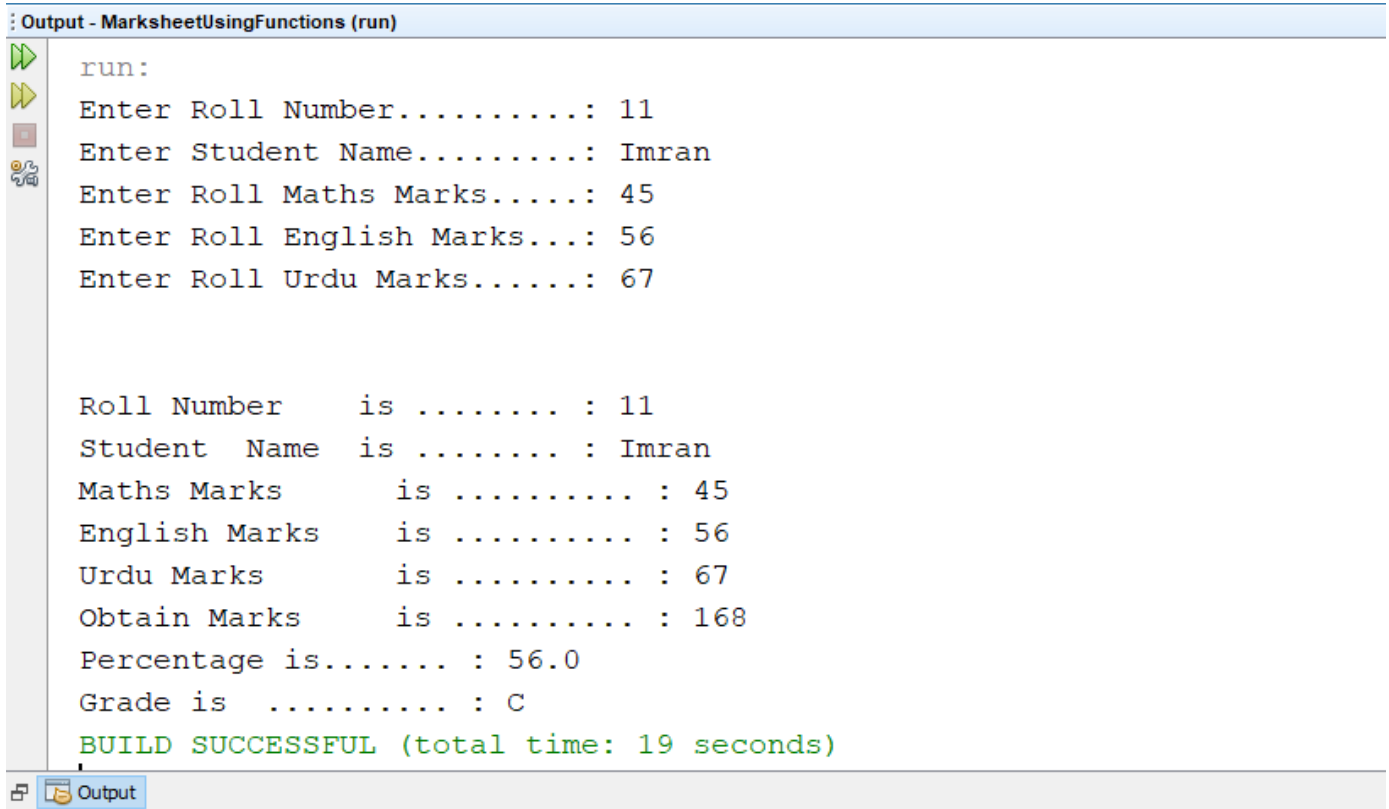
Local variable is declared inside a function whereas Global variable is declared outside the function. Local variables are created when the function has started execution and is lost when the function terminates, on the other hand, Global variable is created as execution starts and is lost when the program ends.

A global variable is one that is accessible to all parts of a program and is usually declared as part of the first lines of code. Java doesn't technically support global variables. As a pure object-oriented language, everything needs to be part of a class.

So above example all variable like (Roll, Name, Math and son on) declare outside the function therefore that are global variable but java language has not support global variables here that are variable are called data member variable of class.

Recursion:

A function that calls itself again and again until some test pattern remain true, is called "**Recursive** "function and this process is known as "**Recursion**".

Example

package testrecursion;

import java.util.Scanner;

public class TestRecursion
{   static int add (int i)
    {   int n;
        if (i==0)
            return 0;
        else
            n = i+ add(i-1);
                return n;
    }

    public static void main(String[] args)
    {

        int x;
        Scanner input = new Scanner(**System**.in);
         System.**out**.print(" Enter any Number: ");
        x= input.nextInt();
        System.**out**.println(" \n Sum : "+ add(x));

    }
}

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help          Search (Ctrl+I)

<default config>

Projects X  Files  Services      Start Page X  TestRecursion.java X
TestRecursion                     Source  History
  Source Packages                  1    package testrecursion;
    testrecursion                   2
      TestRecursion.java            3  import java.util.Scanner;
  Test Packages                     4
  Libraries                         5    public class TestRecursion
  Test Libraries                    6        {      static int add (int i)
                                     7  {            int n;
                                     8          if  (i==0)
                                     9              return 0;
                                    10          else
                                    11              n = i+ add(i-1);
                                    12                  return n;
                                    13          }
                                    14
                                    15
                                    16    public static void main(String[] args)
                                    17        {
                                    18
                                    19          int x;
TestRecursion - Navigator X        20          Scanner input = new Scanner(System.in);
Members    <empty>                 20          System.out.print(" Enter any Number: " );
TestRecursion                      21          x= input.nextInt();
  add(int i) : int                 22          System.out.println(" \n Sum : "+ add(x));
  main(String[] args)              23
                                    24        }
                                    25
                                    26    }

Output - TestRecursion (run)                                                X
   run:
    Enter any Number: 12

    Sum :  78
    BUILD SUCCESSFUL (total time: 10 seconds)

Output                                                             6:1    INS
```
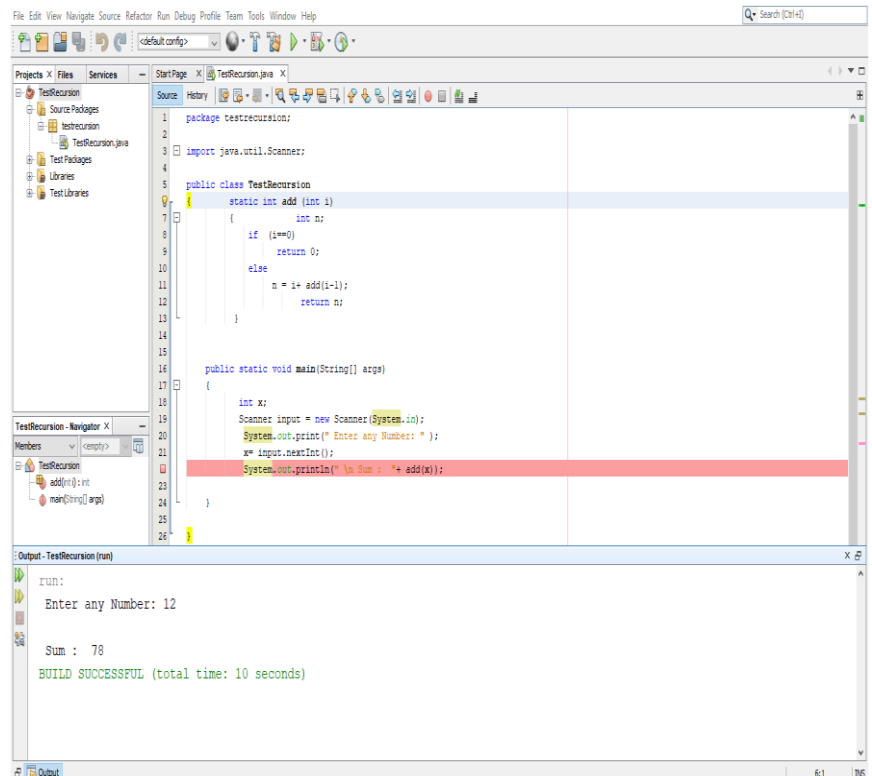
Exercise

Theory Questions

1. Write four advantages of using functions or methods.
2. Explain why some functions do not have a return statement.
3. Explain the different between a local variable and global variable.
4. Describe the recursion?
5. What is difference between the users-define and pre-define functions in Java.

Practical Questions

1. Write a program to display menu are as following. Using by function or methods

    Hint: If user press 1: Liner series

    If user press 2: Odd Number

    If user press 3: Even Series

    If user press 4: exit this program

2. Write a program to find;

    a) Surface area (A=4 pi r 2)   **b)** Area of rectangle (A= width x Length).

3. Write a function to calculate if a number is prime, Return 1 if it is prime and 0 if it is not a prime.
4. Write a program to calculate factorial of a number input by the user using by Recursion.
5. Make a calculator using functions **sum( ), sub( ), mul( ), div( )** for calculations.
6. Write a function **pow**(x,y) to calculate the value of x raised to y.

Objective MCQ's

1. A variable that is declared outside a function is called a _____ variable.
    a) Local
    b) Global
    c) Program
    d) Class
2. A local variable must be declared _____.
    a) Before a function
    b) After a function
    c) Within the braces of a function definition.
    d) With the local keyword.
3. A (n) _____ allows you to treat a related group of Java commands as a single unit.
    a) Statement
    b) Variable
    c) Event
    d) Function

4. When create user-define function should be use _____ with function name.

    a) function
    b) function return type and function parameters (if any)
    c) procedure
    d) nothing all

5. The function calls itself this process is called_____.
   a) Repetitive
   b) Decision
   c) Recursion
   d) loop

6. A program module in Java is called _____.
   a) Variable
   b) Function/method
   c) Sub-routine
   d) Event

7. Function is invoked with a _____.
   a) Return to function.
   b) Function Definition.
   c) Function Call.
   d) Function Prototype.

8. A global variable is defined in a declaration.
   a) In main () only
   b) In the first function that uses it
   c) Outside the any function
   d) Global variable java not supported

9. A variable that is known only with in the function in which it is defined is called a_____.
   a) Global variable
   b) Function
   c) Local variable
   d) Variable scope

10. Which of these are valid reasons for using functions?
    a) They use less memory than repeating the same code.
    b) They keep different program activities separate.
    c) They keep variables safe from other parts of the program.
    d) All option is valid