**Q.1 Write a program to implement Nested,Inner Class & Anonymous Classes.**

class Outer

{

public void displayOuter()

{

System.out.println("I am from Outer Class");

Nested n=new Nested();

n.displayNested();

}

class Nested

{

public void displayNested()

{

System.out.println("I am from Nested Class");

Inner n=new Inner();

n.displayInner();

}

class Inner

{

public void displayInner()

{

System.out.println("I am from Inner Class");

}

}

}

}

interface Anonymous

{

void displayAnonymous();

}

class AllNestedClassesDemo

{

public static void main(String args[])

{

Outer o=new Outer();

o.displayOuter();

Anonymous a=new Anonymous()

{

public void displayAnonymous()

{

System.out.println("I am from Anonymous Class");

}

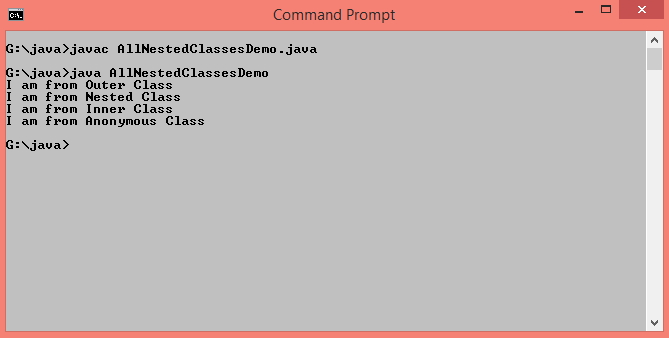
};

a.displayAnonymous();

}

}

**Output :-**



**Q.2 Write a program to implement Abstract class.**

abstract class Vehicle

{

abstract void bike(String name);

abstract void driveBike();

}

class AbstractDemo extends Vehicle

{

void bike(String n)

{

System.out.println(" My bike name is "+n);

}

void driveBike()

{

System.out.println(" I am driving bike in pune.");

}

public static void main(String args[])

{

AbstractDemo a=new AbstractDemo();

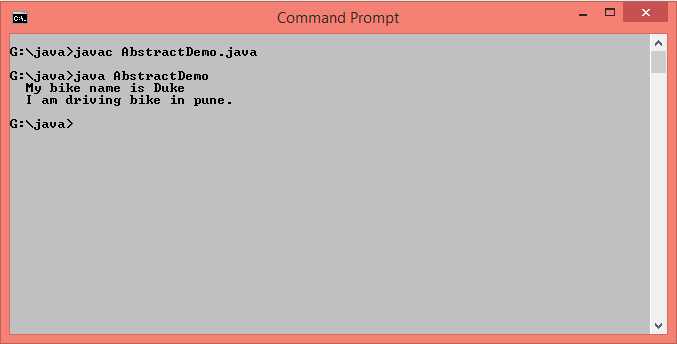
a.bike("Duke");

a.driveBike();

}

}

**Output :-**

****

**Q.3 Write a program to define interface “CalculateArea” with method “area()”. Design class to implement interface.**

interface CalculateArea

{

public void area();

}

class InterfaceDemo implements CalculateArea

{

int radius;

InterfaceDemo(int r)

{

radius=r;

}

public void area()

{

System.out.println("Area of circle= "+(3.14\*radius\*radius));

}

public static void main(String args[])

{

InterfaceDemo i=new InterfaceDemo(5);

i.area();

}

}

**Output :-**

****

**Q.4 Write a program to overload method add() for addition of 2 integers, 3 float values.**

class MethodOverloading

{

void add(int num1,int num2)

{

System.out.println("Addition of two integer values = "+(num1+num2));

}

void add(double num1,double num2,double num3)

{

System.out.println("Addition of three float values = "+(num1+num2+num3));

}

public static void main(String args[])

{

MethodOverloading m=new MethodOverloading();

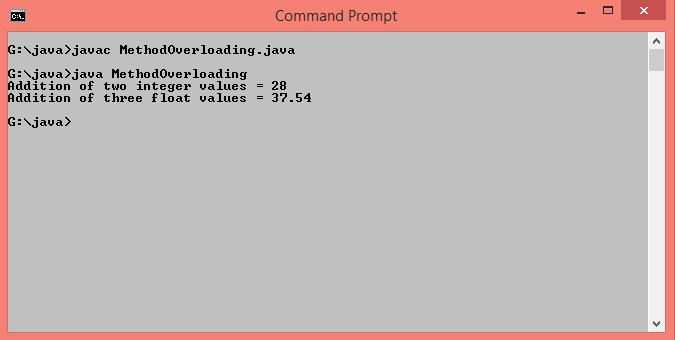
m.add(11,17);

m.add(10.20,12.24,15.10);

}

}

**Output :-**



**Q.5 Write a recursive method for following.**

**a. Write a recursive function to calculate factorial of given number.**

class Factorial

{

static int calculate(int n)

{

if(n==0)

return 1;

else

return(n\*calculate(n-1));

}

public static void main(String args[])

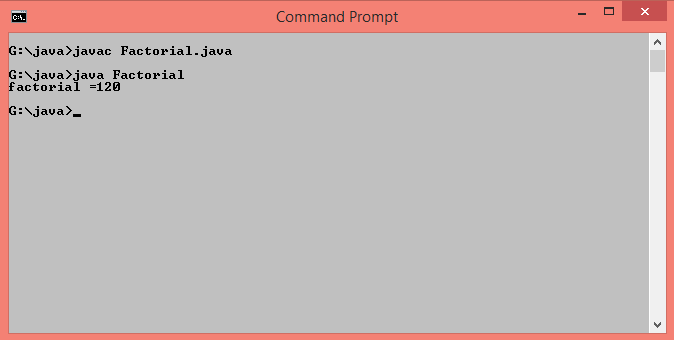
{

System.out.println("factorial ="+calculate(5));

}

}

**Output :-**

****

**b. Write a recursive function to display first 10 terms of Fibonacci series.**

class Fibonacci

{

int a=0,b=1,c,n=1;

Fibonacci()

{

System.out.print(a+" ");

System.out.print(b+" ");

}

void show()

{

if(n<=8)

{

c=a+b;

System.out.print(c+" ");

a=b;

b=c;

n++;

show();

}

}

public static void main(String args[])

{

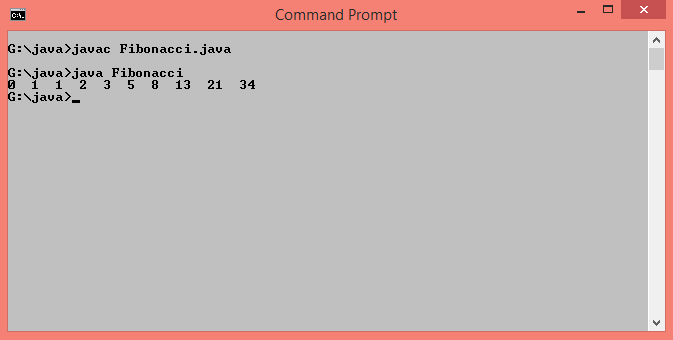
Fibonacci f=new Fibonacci();

f.show();

}

}

**Output :-**



**c. Write a recursive function to calculate sum of all numbers between 5-10.**

class RecursiveSum

{

int sum=0;

int n=5;

void calculate()

{

if(n<=10)

{

sum=sum+n;

n++;

calculate();

}

else

System.out.println("Sum of 5 to 10 Numbers is "+sum);

}

public static void main(String args[])

{

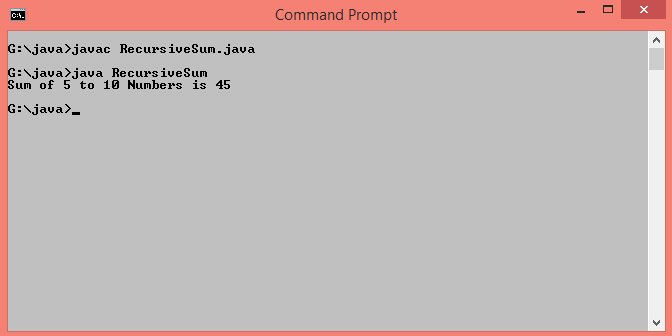
RecursiveSum s=new RecursiveSum();

s.calculate();

}

}

**Output :-**



**d. Write a recursive function to sort list of 10 numbers in ascending order.**

public class RecursiveSort

{

static void sort(int arr[], int n)

{

if (n == 1)

return;

for (int i=0; i<n-1; i++)

if (arr[i] > arr[i+1])

{

int temp = arr[i];

arr[i] = arr[i+1];

arr[i+1] = temp;

}

sort(arr, n-1);

}

public static void main(String[] args)

{

int arr[] = {94, 34, 22, 10, 44, 8, 88,54,87,30};

System.out.println("Array elements before sorting");

for(int i=0;i<arr.length;i++)

{

System.out.print(arr[i]+" ");

}

sort(arr, arr.length);

System.out.println("\nArray elements after sorting");

for(int i=0;i<arr.length;i++)

{

System.out.print(arr[i]+" ");

}

}

}

**Output :-**

