Q.N.1) Design a positive random number generator using method overloading.

Return a random number between 0 to 100 if no parameter is passed by the user.

Return a random number between o to "n", if user pass 1 "n" int number.

Return a random number between 101 to 500 and append a message if user pass the single string message.

Return a random number between the given two numbers, if the user passes two int number.

Return a random number between the given two numbers and append a message if user pass two int number and a string message.

You can use your own logic to calculate random number OR use built in library function.

Ans.

import java.util.Random;

class RandomNumber

{

public void disp(String a)

{

System.out.println("The random Number is ");

}

public void disp(int c)

{

System.out.println(c );

}

}

class Sample2 extends RandomNumber

{

public static void main(String args[])

{

Random r = new Random();

int ran = r.nextInt(100);

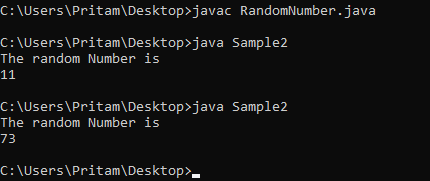
Sample2 s = new Sample2();

s.disp("a");

s.disp(ran);

}}

**OUTPUT**

****

Ans.

import java.util.Random;

import java.util.Scanner;

class RandomNumber

{public void disp(String a)

{System.out.println("The random Number is ");

}

public void disp(int c)

{System.out.println(c );

}}

class Sample2 extends RandomNumber

{

public static void main(String[] args)

{

int b;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number=");

b = sc.nextInt();

sc.close();

Random r = new Random();

int ran = r.nextInt(b);

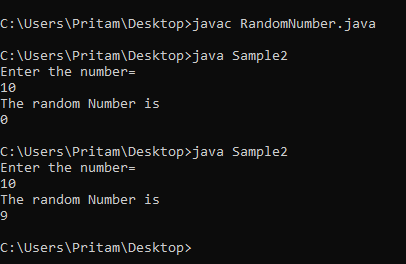
Sample2 s = new Sample2();

s.disp("a");

s.disp(ran);

}

} **OUTPUT**



Ans.

import java.util.Random;

import java.util.Scanner;

class RandomNumber

{public void disp(String a)

{System.out.println("The random Number between 101 and 500 is ");

}

public void disp(int c)

{System.out.println(c );

}}

class Sample2 extends RandomNumber

{

public static void main(String[] args)

{

String b;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the String=");

b = sc.next();

sc.close();

Random r = new Random();

int ran = r.nextInt(500-101)+101;

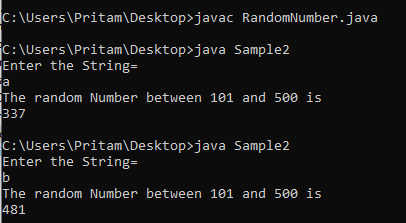
Sample2 s = new Sample2();

s.disp("a");

s.disp(ran);

}

}  **OUTPUT**



Ans.

import java.util.Random;

import java.util.Scanner;

class RandomNumber

{public void disp(String a)

{System.out.println("The random Number is= ");

}

public void disp(int c)

{System.out.println(c );

}}

class Sample2 extends RandomNumber

{

public static void main(String[] args)

{

int min, max;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the First Num=");

min = sc.nextInt();

System.out.println("Enter the Second Num=");

max = sc.nextInt();

sc.close();

Random r = new Random();

int ran = r.nextInt(max-min)+min;

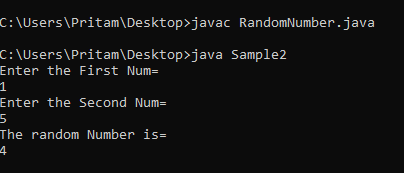
Sample2 s = new Sample2();

s.disp("a");

s.disp(ran);

}

} **OUTPUT**



Q.N.2) Design "Interest calculator" using OOP Concepts. Calculate simple interest and compound interest.

* 1. Explain implementation  of abstraction concept in solution.
  2. Explain implementation  of method overriding concept in solution.

Ans.

abstract class Simple {

double interest,r=2,t=3,Sinterest,p=100;

abstract void cal();

}

class Compound extends Simple {

void cal()

{

interest= p\*Math.pow((1+r/100),t);

System.out.println("Compound int "+ interest);

}

}

class Sinterest extends Simple{

void cal()

{

Sinterest=(p\*t\*r)/100;

System.out.println("Simple int "+ Sinterest);

}

public static void main(String args[])

{

Sinterest a = new Sinterest();

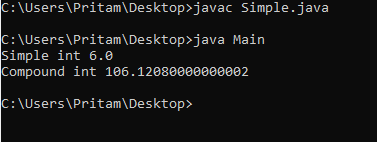
a.cal();

Compound b = new Compound();

b.cal();

}

} **OUTPUT**



Q.N.3) Design a "Property tax calculation system". A city can have buildings like residential, commercial, industrial buildings. All buildings have its own properties like address, total space area, population and valuation. Now, each type of building has different method to calculate the tax.  Design a system which calculates yearly property tax of given building. [Use your own formula for Tax calculation]

* 1. Explain implementation of abstraction concept in solution.
  2. Explain implementation  of method overriding concept in solution.

Ans.

import java.util.\*;

public abstract class City {

abstract void cal();

double property, tax;

public void city(double p) {

this.property = p;

}

}

class Residential extends City{

void cal() {

if(property<=20000) {

System.out.println("No Tax Applicable");

}

else if(property>20000 && property<80000) {

tax = property \* 0.05;

System.out.println("The Tax applied 5% is =" +tax);

}

else {

tax = property \* 0.1;

System.out.println("The Tax applied 10% is =" +tax);

}

}

}

class Commercial extends City{

void cal() {

if(property<=25000) {

System.out.println("No Tax Applicable");

}

else if(property>25000 && property<=80000) {

tax = property \* 0.11;

System.out.println("The Tax applied 11% is =" +tax);

}

else {

tax = property \* 0.18;

System.out.println("The Tax applied 18% is =" +tax);

}

}

}

class Industrial extends City{

void cal() {

if(property<=30000) {

System.out.println("No Tax Applicable");

}

else if(property>30000 && property<=100000) {

tax = property \* 0.15;

System.out.println("The Tax applied 15% is =" +tax);

}

else {

tax = property \* 0.2;

System.out.println("The Tax applied 20% is =" +tax);

}

}

}

class Main{

public static void main(String[] args) {

System.out.println("1.Residential");

System.out.println("\n2.Commercial");

System.out.println("\n3.Industrial");

System.out.println("Select Your Property= ");

Scanner s = new Scanner(System.in);

int a = s.nextInt();

Residential z = new Residential();

Commercial y = new Commercial();

Industrial x = new Industrial();

if(a==1) {

System.out.println("Enter the Property Value=");

double property = s.nextDouble();

z.city(property);

z.cal();

}

else if(a==2) {

System.out.println("Enter the Property Value=");

double property = s.nextDouble();

y.city(property);

y.cal();

}

else {

System.out.println("Enter the Property Value=");

double property = s.nextDouble();

x.city(property);

x.cal();

}

}

}  **OUTPUT**

