PROGRAM 1

Write a program to find the sum of the series 1 + 1/ (2\*2) + 1/(3\*3) +1/(4\*4) + ….. + 1/(n\*n)

import java.io.\*;  
  
class Main  
{  
 static float series(float n)  
 {  
 float i;  
 float sum = 0;  
  
 for(i = 1; i<=n; i++)  
 sum += 1/(i\*i);  
  
 return sum;  
 }  
 public static void main(String[] args)  
 {  
 float n = 3;  
 float res = *series*(n);  
 System.*out*.print("The sum of the given series 1/(i\*i) is: "+ res);  
 }  
}

OUTPUT: The sum of the given series 1/(i\*i) is: 1.3611112

PROGRAM 2

Write a Java program for printing Pascal's Triangle ( 5 rows) using nested loops

import java.util.Scanner;  
import java.io.\*;  
  
class Main  
{  
 public int factorial(int i)  
 {  
 if(i == 0)  
 return 1;  
  
 return(i\*factorial(i-1));  
 }  
 public static void main(String[] args)  
 {  
 int n = 4,i,j;  
 Main g = new Main();  
  
 for(i = 0; i<= n; i++)  
 {  
 for(j = 0; j<= n-i; j++)  
 {  
 System.*out*.print(" ");  
 }  
 for(j = 0; j<= i; j++)  
 {  
 System.*out*.print(" "+g.factorial(i)/(g.factorial(i-j)\*g.factorial(j)));  
 }  
 System.*out*.println();  
 }  
 }  
}

OUTPUT:

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

PROGRAM 3

Write a program that accepts three numbers from the user and prints "increasing" if the numbers are in increasing order, "decreasing" if the numbers are in decreasing order, and "Neither increasing or decreasing order" otherwise

import java.util.Scanner;  
  
class Main  
{  
 public static void main(String[] args)  
 {  
 Scanner obj = new Scanner(System.*in*);  
  
 System.*out*.print("Enter the First Number: ");  
 double x = obj.nextDouble();  
 System.*out*.print("Enter the Second Number: ");  
 double y = obj.nextDouble();  
 System.*out*.print("Enter the Third Number: ");  
 double z = obj.nextDouble();  
  
 if(x<y && y<z)  
 System.*out*.print("The Entered Numbers are in Increasing order...");  
 else if (x>y && y>z)  
 System.*out*.print("The Entered Numbers are in Decreasing order...");  
 else  
 System.*out*.print("The Entered Numbers are in Neither Increasing nor Decreasing Order...");  
 }  
}

OUTPUT 1:

Enter the First Number: 2

Enter the Second Number: 4

Enter the Third Number: 6

The Entered Numbers are in Increasing order...

OUTPUT 2:

Enter the First Number: 6

Enter the Second Number: 5

Enter the Third Number: 4

The Entered Numbers are in Decreasing order...

OUTPUT 3:

Enter the First Number: 3

Enter the Second Number: 5

Enter the Third Number: 1

The Entered Numbers are in Neither Increasing nor Decreasing Order...

PROGRAM 4

Create a Java class called Complex with the following details and variables within it as

(i) Real (ii) Imaginary.

Develop a Java program to perform addition and subtraction of two complex numbers by using the method add() and subtract( ) respectively by passing object as parameter and display result using method display(). Initialize the real and imaginary values of the

complex number using parameterized constructor

import java.util.Scanner;  
  
class Complex  
{  
 int real,imaginary;  
 Complex()  
 {}  
 Complex(int tempReal, int tempImaginary)  
 {  
 real = tempReal;  
 imaginary = tempImaginary;  
 }  
 Complex addComp(Complex C1, Complex C2)  
 {  
 Complex temp = new Complex();  
 temp.real = C1.real + C2.real;  
 temp.imaginary = C1.imaginary + C2.imaginary;  
 return temp;  
 }  
 Complex subtractComp(Complex C1, Complex C2)  
 {  
 Complex temp = new Complex();  
 temp.real = C1.real - C2.real;  
 temp.imaginary = C1.imaginary - C2.imaginary;  
 return temp;  
 }  
 void printComplexNumber()  
 {  
 System.*out*.println(real+"+i"+imaginary);  
 }  
}  
  
class Main  
{  
 public static void main(String[] args)  
 {  
 Complex C1 = new Complex(3,2);  
 System.*out*.print("The Complex Number 1: ");  
 C1.printComplexNumber();  
 Complex C2 = new Complex(9,5);  
 System.*out*.print("The Complex Number 2: ");  
 C2.printComplexNumber();  
  
 Complex C3 = new Complex();  
 C3 = C3.addComp(C1, C2);  
 System.*out*.print("Sum of the Two Complex Numbers are: ");  
 C3.printComplexNumber();  
 C3 = C3.subtractComp(C1,C2);  
 System.*out*.print("Difference of the Two Complex Numbers are: ");  
 C3.printComplexNumber();  
 }  
}

OUTPUT:

The Complex Number 1: 3+i2

The Complex Number 2: 9+i5

Sum of the Two Complex Numbers are: 12+i7

Difference of the Two Complex Numbers are: -6+i-3

PROGRAM 5

A class called MyTime, which models a time instance with private instance variables: hour: between 0 to 23, minute: between 0 to 59, constructor shall invoke the setTime() method to set the instance variable (setTime(int hour, int minute): It shall check if the given hour and minute are valid before setting the instance variables). Define methods - getHour(), getMinute(),nextMinute( )Update this instance to the next minute and return this instance. Take note that the nextMinute() of 23:59 is 00:00 nextHour() is similar to the above.

Write the code for the MyTime class. Also write a test program (called TestMyTime) to test all the methods defined in the MyTime class

class MyTime  
{  
 private int hour,minute;  
 MyTime(int hr,int min)  
 {  
 SetTime(hr,min);  
 }  
 void SetTime(int hr,int min)  
 {  
 if(hr>=0 && hr<=23 && min >=0 && min <=59)  
 {  
 hour = hr;  
 minute = min;  
 }  
 else  
 System.*out*.println("Invalid Time");  
 }  
 int getHour()  
 {  
 return hour;  
 }  
 int getMinute()  
 {  
 return minute;  
 }  
 void nextHour()  
 {  
 if(hour>23)  
 {  
 hour = 0;  
 }  
  
 }  
 void nextMinute()  
 {  
 minute++;  
 if(minute>59)  
 {  
 minute = 0;  
 hour++;  
 if(hour > 23)  
 {  
 hour = 0;  
 }  
 }  
 }  
}  
  
public class Main  
{  
 public static void main(String[] args)  
 {  
 MyTime obj = new MyTime(23,59);  
 System.*out*.println("Hour- "+obj.getHour());  
 System.*out*.println("Minute- "+obj.getMinute());  
  
 obj.nextHour();  
 obj.nextMinute();  
  
 System.*out*.println("Updated Hour and Minute");  
 System.*out*.println("Hour- "+obj.getHour());  
 System.*out*.println("Minute- "+obj.getMinute());  
  
 }  
}

OUTPUT:

Hour- 23

Minute- 59

Updated Hour and Minute

Hour- 0

Minute- 0

PROGRAM 6

Assume that a bank maintains two kinds of accounts for customers one called as savings account and the other as current account. Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct and Sav-acct to

make them more specific to their requirements. The savings account provides compound interest and withdrawal facilities. The current account does not provide interest. Current account holders should also maintain a minimum balance (Rs 5000) and if the balance falls below this level, a service charge (Rs 100) is imposed. Include the necessary methods in order to achieve the following tasks:

Accept deposit from customer and update the balance.

Display the balance.

Compute and deposit interest

Permit withdrawal and update the balance

Check for the minimum balance(only for Current account), impose

penalty if necessary and update the balance.

import java.util.\*;  
import java.lang.\*;  
class Account  
{  
 String acc\_name;  
 double acc\_no;  
 int acc\_type;  
 double balance;  
 public void getdata(String name,double no,int type,double bal)  
 {  
 acc\_name=name;  
 acc\_no=no;  
 acc\_type=type;  
 balance=bal;  
 }  
}  
class Savings extends Account  
{  
 public void deposit(double amt)  
 {  
 balance = balance+amt;  
 System.*out*.println(balance);  
 }  
 public void withdraw(double amt)  
 {  
 balance = balance-amt;  
 System.*out*.println(balance);  
 }  
 void interest(int time,int no)  
 {  
 double intr = balance\*(1+6/no);  
 intr=Math.*pow*(intr,(time\*no));  
 System.*out*.println("Intertest calculated for the given amount is: "+intr);  
 balance += intr;  
 System.*out*.print("The new balance after intrest addition is: "+balance);  
 }  
}  
class Current extends Account  
{  
 public void deposit(double amt)  
 {  
 balance=balance+amt;  
 System.*out*.println(balance);  
 }  
 public void withdraw(double amt)  
 {  
 balance=balance-amt;  
 System.*out*.println(balance);  
 check(balance);  
 }  
 public void check(double amt)  
 {  
 if(balance<10000)  
 {  
 balance =balance-500;  
 System.*out*.println("Insufficient Balance, Hence balance after Penalty Deduction: "+balance);  
 }  
 }  
}  
class Main  
{  
 public static void main(String args[])  
 {  
 Scanner sc = new Scanner(System.*in*);  
 int temp = 1;  
 while (temp == 1)  
 {  
 double amt = 0;  
 System.*out*.print("Enter the Name: ");  
 sc.next();  
 String name = sc.nextLine();  
 System.*out*.print(name);  
 System.*out*.print("Enter account number: ");  
 double no = sc.nextDouble();  
 System.*out*.print("Account type\n0:Savings\n1:Current\n\nEnter your choice here: ");  
 int type = sc.nextInt();  
 System.*out*.print("Enter the Amount: ");  
 amt = sc.nextDouble();  
  
 if (type == 0)  
 {  
 Savings s = new Savings();  
 s.getdata(name, no, type, amt);  
 System.*out*.print("\n1.Deposit\n2.Withdraw\n3.Interest\n\n Enter your choice here: ");  
 int temp3 = sc.nextInt();  
  
 if (temp3 == 1)  
 {  
 System.*out*.print("Enter Amount to be deposited: ");  
 double amt1 = sc.nextDouble();  
 System.*out*.print("Final Amount: ");  
 s.deposit(amt1);  
 }  
 else if (temp3 == 2)  
 {  
 System.*out*.print("Enter Amoumt to be withdrawn: ");  
 double amt1 = sc.nextDouble();  
 System.*out*.print("Final Amount: ");  
 s.withdraw(amt1);  
 }  
 else if (temp3 == 3)  
 {  
 System.*out*.print("Enter time period");  
 int tp = sc.nextInt();  
 System.*out*.print("Enter no of times");  
 int nof = sc.nextInt();  
 System.*out*.print("Final Amount: ");  
 s.interest(tp, nof);  
 }  
 }  
 else if (type == 1)  
 {  
 Current c = new Current();  
 c.getdata(name, no, type, amt);  
 System.*out*.print("\n1.Deposit\n\2.Withdraw\n\n Enter your choice here: ");  
 int temp3 = sc.nextInt();  
  
 if (temp3 == 1)  
 {  
 System.*out*.print("Enter Amoumt: ");  
 double amt1 = sc.nextDouble();  
 System.*out*.print("Final Amount: ");  
 c.deposit(amt1);  
 }  
 else if (temp3 == 2)  
 {  
 System.*out*.print("Enter Amoumt to be Withdrawn: ");  
 double amt1 = sc.nextDouble();  
 System.*out*.print("Amount after withdrawl: ");  
 c.withdraw(amt1);  
 }  
 }  
 System.*out*.println("\nTo continue 1 else 0: ");  
 temp = sc.nextInt();  
 }  
 }  
}

OUTPUT 1:

Enter the Name: Hariharan

Enter account number: 2021

Account type

0:Savings

1:Current

Enter your choice here: 0

Enter the Amount: 25000

1.Deposit

2.Withdraw

3.Interest

Enter your choice here: 1

Enter Amount to be deposited: 5000

Final Amount: 30000.0

OUTPUT 2:

Enter the Name: Hariharan

Enter account number: 2021

Account type

0:Savings

1:Current

Enter your choice here: 0

Enter the Amount: 25000

1.Deposit

2.Withdraw

3.Interest

Enter your choice here: 2

Enter Amount to be withdrawn: 20000

Final Amount: 5000.0

OUTPUT 3:

Enter the Name: Hariharan

Enter account number: 2021

Account type

0:Savings

1:Current

Enter your choice here: 0

Enter the Amount: 25000

1.Deposit

2.Withdraw

3.Interest

Enter your choice here: 3

Enter time period:3

Enter no of times:2

Final Amount: Intrest calculated for the given amount is: 1.0E30

The new balance after intrest addition is: 1.0E30

OUTPUT 4:

Enter the Name: Hariharan

Enter account number: 2021

Account type

0:Savings

1:Current

Enter your choice here: 1

Enter the Amount: 25000

1.Deposit

.Withdraw

Enter your choice here: 1

Enter Amoumt: 5000

Final Amount: 30000.0

OUTPUT 5:

Enter the Name: Hariharan

Enter account number: 2021

Account type

0:Savings

1:Current

Enter your choice here: 1

Enter the Amount: 25000

1.Deposit

.Withdraw

Enter your choice here: 2

Enter Amoumt to be Withdrawn: 20000

Amount after withdrawl: 5000.0

Insufficient Balance, Hence balance after Penalty Deduction: 4500.0

PROGRAM 7

Design a base class Circle with member variables (radius of type double and color of type character), methods (getRadius(), getArea()) and constructors (Circle(radius), Circle(radius, color)). Derive subclass called Cylinder from the super class Circle with member variable (height) of type double, public methods (getHeight(), getVolume(), getArea()) and constructors(Cylinder(height),Cylinder(height,radius),Cylinder(height, radius, color)). Create the two instances of cylinder and print similar cylinders if the area, volume and color of cylinders are same. Demonstrate the code reuse and polymorphism properties of Object

oriented programming by inheriting the constructors and methods of the base class.

Derive subclass called Cylinder from the superclass Circle with member variable (height) of type double, public methods (getHeight(), getVolume(), getArea()) and its constructors (Cylinder(height, radius), Cylinder(height, radius,color)). Create the two instances of cylinder and print similar cylinders if the area, volume and color of cylinders are same. Demonstrate the code reuse and polymorphism properties of Object oriented programming by inheriting the constructors and methods of the base class.

import java.util.Scanner;  
  
class circle  
{  
 double radius;  
 String color;  
 circle()  
 {  
 radius=1.0;  
 color="blue";  
 }  
 circle(double radius)  
 {  
 this.radius=radius;  
 color="blue";  
 }  
 circle(double radius,String color)  
 {  
 this.radius=radius;  
 this.color=color;  
 }  
 double getarea()  
 {  
 return Math.*PI*\*radius\*radius;  
 }  
 double getradius()  
 {  
 return radius;  
 }  
 String getcolor(){return color;}  
}  
class cylinder extends circle{  
 double height;  
  
 double getheight()  
 { return height;  
 }  
 cylinder()  
 {  
 super();  
 height=2.0;  
 }  
 cylinder(double height)  
 {  
 super();  
 this.height=height;  
  
 }  
 cylinder(double height, double radius)  
 {  
 super(radius);  
 this.height=height;  
 }  
 cylinder(double height,double radius, String color)  
 {  
 super(radius,color);  
 this.height=height;  
 }  
 double getarea()  
 {  
 return ((2\* Math.*PI*\*radius\*height)+(2\* Math.*PI*\*radius\*radius));  
 }  
 double getvolume()  
 {  
 return (super.getarea()\*height);  
 }  
 void display()  
 {  
 System.*out*.println("\nRadius is "+super.radius+",Height is "+height+", Color is "+super.color+",Area is "+getarea()+",Volume is "+getvolume());  
 }  
 void check (cylinder c1,cylinder c2,int i,int j){  
 if((c1.radius==c2.radius)&& (c1.height==c2.height)&&(c1.color.equalsIgnoreCase(c2.color)))  
 System.*out*.println("The cylinders "+(i+1)+" and "+(j+1)+"are similar");  
 }  
  
}  
public class Main {  
 public static void main(String[] args) {  
 Scanner s = new Scanner(System.*in*);  
 cylinder[] c = new cylinder[4];  
 int i;  
 c[0] = new cylinder();  
 c[1] = new cylinder(3.0);  
 c[2] = new cylinder(3.0, 4.0, "Green");  
 System.*out*.println("Enter the details of cylinder 4 (height , radius and color)");  
 double h = s.nextDouble();  
 double r = s.nextDouble();  
 s.nextLine();  
 String st = s.nextLine();  
 c[3] = new cylinder(h, r, st);  
 for (i = 0; i < 4; i++) {  
 System.*out*.println("The dimensions of cylinder " + (i + 1) + " is ");  
 c[i].display();  
  
 }  
 for (i = 0; i < 4; i++) {  
 int j;  
 for (j = i + 1; j < 4; j++) {  
 c[i].check(c[i], c[j], i, j);  
 }  
 }  
  
  
 }  
}

OUTPUT:

Enter the details of cylinder 4 (height , radius and color)

1 2 RED

The dimensions of cylinder 1 is

Radius is 1.0,Height is 2.0, Color is blue,Area is 18.84955592153876,Volume is 6.283185307179586

The dimensions of cylinder 2 is

Radius is 1.0,Height is 3.0, Color is blue,Area is 25.132741228718345,Volume is 9.42477796076938

The dimensions of cylinder 3 is

Radius is 4.0,Height is 3.0, Color is Green,Area is 175.92918860102841,Volume is 150.79644737231007

The dimensions of cylinder 4 is

Radius is 2.0,Height is 1.0, Color is ,Area is 37.69911184307752,Volume is 12.566370614359172

PROGRAM 8

Create two classes called HDFC Account and State Bank Account. That implements all the methods defined in interface Account. Declare the methods getBalance, deposit and withdraw in Account interface. HDFC Account uses member variables deposits and withdrawals for

maintaining the balance, where as State Bank Account uses only balance to maintain the balance. In the main method create objects of HDFC Account and State Bank Account, but assigned them to the reference of the interface Account. Also write an method to print balance in main which prints the balance amount

class TestAccountInterface  
{  
 public static void main(String s[])  
 {  
 IAccount account = new HDFCAccount();  
 System.*out*.println("Transacting using HDFC Account");  
 *transactOnAccount*(account);  
 System.*out*.println();  
 account = new StateBankAccount();  
 System.*out*.println("Transacting using State Bank Account");  
 *transactOnAccount*(account);  
 }  
 public static void transactOnAccount(IAccount account)  
 {  
 System.*out*.println("------------------------------");  
 account.deposit(10000.0);  
 *printBalance*("depositing 10,000.0", account);  
 account.withdraw(2500.0);  
 *printBalance*("withdrawing 2,500.0", account);  
 account.withdraw(4100.0);  
 *printBalance*("withdrawing 4,100.0", account);  
 account.deposit(5000.0);  
 *printBalance*("depositing 5,000.0", account);  
 System.*out*.println("------------------------------");  
 }  
 public static void printBalance(String message, IAccount account)  
 {  
 System.*out*.println("The balance after " + message + " is " + account.getBalance() +".");  
 }  
}  
interface IAccount  
{  
 double getBalance();  
 void deposit(double amount);  
 void withdraw(double amount);  
}  
class HDFCAccount implements IAccount  
{  
 double deposits;  
 double withdrawals;  
 public double getBalance()  
 {  
 return deposits - withdrawals;  
 }  
 public void deposit(double amount)  
 {  
 deposits += amount;  
 }  
 public void withdraw(double amount)  
 {  
 withdrawals += amount;  
 }  
}  
class StateBankAccount implements IAccount  
{  
 double balance;  
 public double getBalance()  
 {  
 return balance;  
 }  
 public void deposit(double amount)  
 {  
 balance += amount;  
 }  
 public void withdraw(double amount)  
 {  
 balance -= amount;  
 }  
}

OUTPUT:

Transacting using HDFC Account

------------------------------

The balance after depositing 10,000.0 is 10000.0.

The balance after withdrawing 2,500.0 is 7500.0.

The balance after withdrawing 4,100.0 is 3400.0.

The balance after depositing 5,000.0 is 8400.0.

------------------------------

Transacting using State Bank Account

------------------------------

The balance after depositing 10,000.0 is 10000.0.

The balance after withdrawing 2,500.0 is 7500.0.

The balance after withdrawing 4,100.0 is 3400.0.

The balance after depositing 5,000.0 is 8400.0.

------------------------------

PROGRAM 9

Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class internals has an array that stores the internal marks scored in six courses of the current semester of the student. Create another package

SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in six courses of the current semester of the student. Import the two packages in a file that declares the final marks of N students in all six courses

package cie;  
import java.util.Scanner;  
  
public class Internals extends Student  
{  
 public int ciemarks[] = new int[6];  
 public void readCieMarks()  
 {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print("Enter the CIE marks of all 6 Subjects: ");  
  
 for (int i = 0; i<6; i++)  
 ciemarks[i] = sc.nextInt();  
 }  
}

package cie;  
import java.util.Scanner;  
  
public class Student  
{  
 public String name,usn;  
 int sem;  
 public void resdStudent()  
 {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print("Enter the Name, USN and the Semester of the Student: ");  
 name = sc.next();  
 usn = sc.next();  
 sem = sc.nextInt();  
 }  
}

package see;  
import java.util.Scanner;  
import cie.\*;  
  
public class Externals extends Internals  
{  
 public int seemarks[] = new int[6];  
 public void readSeeMarks()  
 {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print("Enter the SEE marks of all 6 Subjects: ");  
  
 for(int i = 0; i<6; i++)  
 {  
 seemarks[i] = sc.nextInt();  
 }  
 }  
}

package exam;  
import java.util.Scanner;  
import cie.\*;  
import see.\*;  
  
public class Result  
{  
 public static void main(String[] args)  
 {  
 int n;  
 Scanner sc1 = new Scanner(System.*in*);  
 System.*out*.print("Enter the Number of Students: ");  
 n = sc1.nextInt();  
 Externals e[] = new Externals[n];  
 int totalmarks[] = new int[6];  
  
 for(int j = 0; j<n; j++)  
 {  
 System.*out*.print("\nEnter the Student "+(j+1)+" details\n ");  
 e[j] = new Externals();  
 e[j].resdStudent();  
 e[j].readCieMarks();  
 e[j].readSeeMarks();  
 for(int i = 0; i<6; i++)  
 totalmarks[i] = e[j].ciemarks[i]+e[j].seemarks[i];  
 System.*out*.print("\nTotal Marks of "+e[j].name+" in the 6 subjects: ");  
  
 for(int i =0; i<6; i++)  
 System.*out*.print("\t"+totalmarks[i]);  
 System.*out*.println();  
 }  
 System.*out*.println("\nGOOD BYEEE............");  
 }  
}

OUTPUT:

Enter the Number of Students: 2

Enter the Student 1 details

Enter the Name, USN and the Semester of the Student: random1 21is 3

Enter the CIE marks of all 6 Subjects: 40 40 40 40 40 40

Enter the SEE marks of all 6 Subjects: 60 60 60 60 60 60

Total Marks of random1 in the 6 subjects: 100 100 100 100 100 100

Enter the Student 2 details

Enter the Name, USN and the Semester of the Student: random2 21is 3

Enter the CIE marks of all 6 Subjects: 35 35 35 35 35 35

Enter the SEE marks of all 6 Subjects: 65 65 65 65 65 65

Total Marks of random2 in the 6 subjects: 100 100 100 100 100 100

GOOD BYEEE............

PROGRAM 10

Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number

import java.util.\*;  
class EvenNum implements Runnable  
{  
 public int a;  
 public EvenNum(int a)  
 {  
 this.a = a;  
 }  
 public void run()  
 {  
 System.*out*.println("The Thread "+ a +" is EVEN and Square of " + a + " is : " + a \* a);  
 }  
}  
class OddNum implements Runnable  
{  
 public int a;  
 public OddNum(int a)  
 {  
 this.a = a;  
 }  
 public void run()  
 {  
 System.*out*.println("The Thread "+ a +" is ODD and Cube of " + a + " is: " + a \* a \* a);  
 }  
}  
  
class RandomNumGenerator extends Thread  
{  
 public void run()  
 {  
 int n = 0;  
 Random rand = new Random();  
 try  
 {  
 for (int i = 0; i < 10; i++)  
 {  
 n = rand.nextInt(20);  
 System.*out*.println("Generated Number is " + n);  
 if (n % 2 == 0)  
 {  
 Thread thread1 = new Thread(new EvenNum(n));  
 thread1.start();  
 }  
 else  
 {  
 Thread thread2 = new Thread(new OddNum(n));  
 thread2.start();  
 }  
 Thread.*sleep*(1000);  
 System.*out*.println("------------------------------------");  
 }  
 }  
 catch (Exception ex) {  
 System.*out*.println(ex.getMessage());  
 }  
 }  
}  
public class Main  
{  
 public static void main(String[] args)  
 {  
 RandomNumGenerator rand\_num = new RandomNumGenerator();  
 rand\_num.start();  
 }  
}

OUTPUT:

Generated Number is 7

The Thread 7 is ODD and Cube of 7 is: 343

------------------------------------

Generated Number is 12

The Thread 12 is EVEN and Square of 12 is : 144

------------------------------------

Generated Number is 15

The Thread 15 is ODD and Cube of 15 is: 3375

------------------------------------

Generated Number is 17

The Thread 17 is ODD and Cube of 17 is: 4913

------------------------------------

Generated Number is 9

The Thread 9 is ODD and Cube of 9 is: 729

------------------------------------

Generated Number is 13

The Thread 13 is ODD and Cube of 13 is: 2197

------------------------------------

Generated Number is 10

The Thread 10 is EVEN and Square of 10 is : 100

------------------------------------

Generated Number is 6

The Thread 6 is EVEN and Square of 6 is : 36

------------------------------------

Generated Number is 14

The Thread 14 is EVEN and Square of 14 is : 196

------------------------------------

Generated Number is 10

The Thread 10 is EVEN and Square of 10 is : 100

------------------------------------

PROGRAM 11

Write a java program to handle the following exceptions based on choice made by the user by writing suitable try and catch block.

i) Arithmetic Exception

ii) Array Index Out Of Bounds Exception

iii) Number Format Exception

iv) String Index Out Of Bound Exception

v) Null Pointer Exception

import java.util.Scanner;  
class Main  
{  
 public static void main(String[] args)  
 {  
 int op;  
 Scanner sc = new Scanner(System.*in*);  
 do  
 {  
 System.*out*.print("\n\n$$\*\*\*\*\*\*MAIN MENU\*\*\*\*\*\*$$\n");  
 System.*out*.print("\tException Handling Implementation\n");  
 System.*out*.print("1:Arithmetic Exception\n2:Array Index out of bound exception\n3:Number format exception\n4:String index out of bound exception\n5:Null pointer exception\n6:Exit..");  
 System.*out*.print("\n\n$$\*\*\*\*\*\*END OF MAIN MENU\*\*\*\*\*\*$$\n");  
 System.*out*.print("Enter your choice: ");  
 op = sc.nextInt();  
  
 switch (op)  
 {  
 case 1:  
 try  
 {  
 int num1=30, num2=0;  
 int output=num1/num2;  
 System.*out*.println ("Result: "+output);  
 }  
 catch(ArithmeticException e)  
 {  
 System.*out*.println ("You Shouldn't divide a number by zero");  
 System.*out*.print("SYSTEM MESSAGE: "+e);  
 }  
 break;  
 case 2:  
 try  
 {  
 int a[]=new int[10];  
 a[11] = 9;  
 }  
 catch(ArrayIndexOutOfBoundsException e)  
 {  
 System.*out*.println ("ArrayIndexOutOfBounds Exception occurred");  
 System.*out*.println ("System Message: "+e);  
 }  
 break;  
 case 3:  
 try  
 {  
 int num=Integer.*parseInt* ("XYZ") ;  
 System.*out*.println(num);  
 }  
 catch(NumberFormatException e)  
 {  
 System.*out*.println("Number format exception occurred");  
 System.*out*.print("SYSTEM MESSAGE: "+e);  
 }  
 break;  
 case 4:  
 try  
 {  
 String str="beginnersbook";  
 System.*out*.println(str.length());  
 char c = str.charAt(40);  
 System.*out*.println(c);  
 }  
 catch(StringIndexOutOfBoundsException e)  
 {  
 System.*out*.println("StringIndexOutOfBoundsException.");  
 System.*out*.print("SYSTEM MESSAGE: "+e);  
 }  
 break;  
 case 5:  
 try  
 {  
 String str = null;  
 System.*out*.println(str.length());  
 }  
 catch (NullPointerException e)  
 {  
 System.*out*.println("NullPointerException..");  
 System.*out*.print("SYSTEM MESSAGE: "+e);  
 }  
 break;  
 case 6:  
 System.*out*.print("GOOD BYEEE........");  
 break;  
 }  
 }while(op != 6);  
 }  
}

OUTPUT:

$$\*\*\*\*\*\*MAIN MENU\*\*\*\*\*\*$$

Exception Handling Implementation

1:Arithmetic Exception

2:Array Index out of bound exception

3:Number format exception

4:String index out of bound exception

5:Null pointer exception

6:Exit..

$$\*\*\*\*\*\*END OF MAIN MENU\*\*\*\*\*\*$$

Enter your choice: 1

You Shouldn't divide a number by zero

SYSTEM MESSAGE: java.lang.ArithmeticException: / by zero

$$\*\*\*\*\*\*MAIN MENU\*\*\*\*\*\*$$

Exception Handling Implementation

1:Arithmetic Exception

2:Array Index out of bound exception

3:Number format exception

4:String index out of bound exception

5:Null pointer exception

6:Exit..

$$\*\*\*\*\*\*END OF MAIN MENU\*\*\*\*\*\*$$

Enter your choice: 2

ArrayIndexOutOfBounds Exception occurred

System Message: java.lang.ArrayIndexOutOfBoundsException: Index 11 out of bounds for length 10

$$\*\*\*\*\*\*MAIN MENU\*\*\*\*\*\*$$

Exception Handling Implementation

1:Arithmetic Exception

2:Array Index out of bound exception

3:Number format exception

4:String index out of bound exception

5:Null pointer exception

6:Exit..

$$\*\*\*\*\*\*END OF MAIN MENU\*\*\*\*\*\*$$

Enter your choice: 3

Number format exception occurred

SYSTEM MESSAGE: java.lang.NumberFormatException: For input string: "XYZ"

$$\*\*\*\*\*\*MAIN MENU\*\*\*\*\*\*$$

Exception Handling Implementation

1:Arithmetic Exception

2:Array Index out of bound exception

3:Number format exception

4:String index out of bound exception

5:Null pointer exception

6:Exit..

$$\*\*\*\*\*\*END OF MAIN MENU\*\*\*\*\*\*$$

Enter your choice: 4

13

StringIndexOutOfBoundsException.

SYSTEM MESSAGE: java.lang.StringIndexOutOfBoundsException: Index 40 out of bounds for length 13

$$\*\*\*\*\*\*MAIN MENU\*\*\*\*\*\*$$

Exception Handling Implementation

1:Arithmetic Exception

2:Array Index out of bound exception

3:Number format exception

4:String index out of bound exception

5:Null pointer exception

6:Exit..

$$\*\*\*\*\*\*END OF MAIN MENU\*\*\*\*\*\*$$

Enter your choice: 5

NullPointerException..

SYSTEM MESSAGE: java.lang.NullPointerException: Cannot invoke "String.length()" because "str" is null

$$\*\*\*\*\*\*MAIN MENU\*\*\*\*\*\*$$

Exception Handling Implementation

1:Arithmetic Exception

2:Array Index out of bound exception

3:Number format exception

4:String index out of bound exception

5:Null pointer exception

6:Exit..

$$\*\*\*\*\*\*END OF MAIN MENU\*\*\*\*\*\*$$

Enter your choice: 6

GOOD BYEEE........

PROGRAM 12

Define a class Sort with generic method by name Arrange(T[]) and Display(T[]). Write a program to sort array elements of different data types

class Sort  
{  
 void Arrange(int t[])  
 {  
 int pos;  
 int temp;  
 for (int i = 0; i < t.length; i++)  
 {  
 pos = i;  
 for (int j = i+1; j < t.length; j++)  
 {  
 if (t[j] < t[pos])  
 {  
 pos = j;  
 }  
 }  
  
 temp = t[pos];  
 t[pos] = t[i];  
 t[i] = temp;  
 }  
 }  
  
 void display(int t[])   
 {  
 for (int i=0; i<t.length; i++)  
 {  
 System.*out*.print(t[i]+" ");  
 }  
 }  
  
 public static void main(String args[])  
 {  
 Sort ob = new Sort();  
 int arr[] = {64,25,12,22,11};  
 ob.Arrange(arr);  
 ob.display(arr);  
 }  
}

OUTPUT:

11 12 22 25 64