JAVA LAB PROGRAMS

NAME:GANASHREE C M

USN: 1BM22CS097

SECTION:3B

1. QUADRATIC EQUATION USING JAVA

CODE:

import java.util.Scanner;

class Quadratic {

int a, b, c;

double r1, r2, d;

void getd() {

Scanner s = new Scanner(System.in);

System.out.println("Enter the coefficients of a, b, c");

a = s.nextInt();

b = s.nextInt();

c = s.nextInt();

}

void compute() {

if (a == 0) {

System.out.println("Not a quadratic equation");

System.out.println("Enter a non-zero value for a:");

Scanner s = new Scanner(System.in);

a = s.nextInt();

}

d = b \* b - 4 \* a \* c;

if (d == 0) {

r1 = (-b) / (2 \* a);

System.out.println("Roots are real and equal");

System.out.println("Root1 = Root2 = " + r1);

} else if (d > 0) {

r1 = ((-b) + (Math.sqrt(d))) / (2 \* a);

r2 = ((-b) - (Math.sqrt(d))) / (2 \* a);

System.out.println("Roots are real and distinct");

System.out.println("Root1 = " + r1 + " Root2 = " + r2);

} else if (d < 0) {

System.out.println("Roots are imaginary");

r1 = (-b) / (2 \* a);

r2 = Math.sqrt(-d) / (2 \* a);

System.out.println("Root1 = " + r1 + " + i" + r2);

System.out.println("Root1 = " + r1 + " - i" + r2);

}

}

}

class Main {

public static void main(String args[]) {

System.out.println("name Ganashree");

Quadratic q = new Quadratic();

q.getd();

q.compute();

}

} import java.util.Scanner;

class Quadratic {

int a, b, c;

double r1, r2, d;

void getd() {

Scanner s = new Scanner(System.in);

System.out.println("Enter the coefficients of a, b, c");

a = s.nextInt();

b = s.nextInt();

c = s.nextInt();

}

void compute() {

if (a == 0) {

System.out.println("Not a quadratic equation");

System.out.println("Enter a non-zero value for a:");

Scanner s = new Scanner(System.in);

a = s.nextInt();

}

d = b \* b - 4 \* a \* c;

if (d == 0) {

r1 = (-b) / (2 \* a);

System.out.println("Roots are real and equal");

System.out.println("Root1 = Root2 = " + r1);

} else if (d > 0) {

r1 = ((-b) + (Math.sqrt(d))) / (2 \* a);

r2 = ((-b) - (Math.sqrt(d))) / (2 \* a);

System.out.println("Roots are real and distinct");

System.out.println("Root1 = " + r1 + " Root2 = " + r2);

} else if (d < 0) {

System.out.println("Roots are imaginary");

r1 = (-b) / (2 \* a);

r2 = Math.sqrt(-d) / (2 \* a);

System.out.println("Root1 = " + r1 + " + i" + r2);

System.out.println("Root1 = " + r1 + " - i" + r2);

}

}

}

class Main {

public static void main(String args[]) {

System.out.println("name Ganashree");

Quadratic q = new Quadratic();

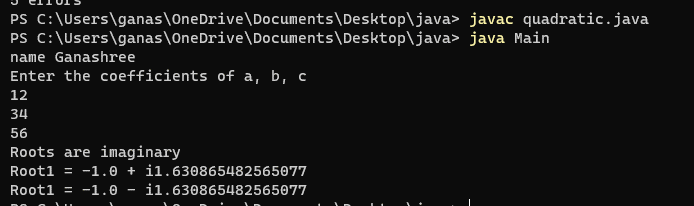
q.getd();

q.compute();

}

}

OUTPUT:



1. Write a Java program to create a class Student with members USN, name, marks(6 subjects). Include methods to accept student details and marks, Also include a method to calculate the percentage and display appropriate details. (Array of student object to be created)

CODE:

import java.util.Scanner;

class Student{

private int usn;

private String name;

private int[] marks;

Student(int usn,String name)

{

this.usn=usn;

this.name=name;

this.marks=new int[6];

}

public void accept()

{

Scanner input=new Scanner(System.in);

System.out.println("enter the name");

this.name=input.next();

System.out.println("enter the usn");

this.usn=input.nextInt();

System.out.println("enter the marks of all subject");

for(int i=0;i<6;i++)

{

System.out.println("enter the marks of "+(i+1)+" subject");

this.marks[i]=input.nextInt();

}

}

public double calcpercentage()

{

int totalmarks=0;

for(int i=0;i<6;i++)

{

totalmarks=totalmarks+marks[i];

}

double percentage=totalmarks/6;

return percentage;

}

void display()

{

System.out.println("the name is"+this.name);

System.out.println("the usn is"+this.usn);

System.out.println("the percentage is "+calcpercentage()+"%");

}

}

class main

{

public static void main(String args[])

{

System.out.println(“Ganashree lab 1");

System.out.println("enter the number of students");

Scanner input=new Scanner(System.in);

int num=input.nextInt();

Student[] s=new Student[num];

for(int i=0;i<num;i++)

{

System.out.println("enter the details of "+(i+1)+" student");

s[i]=new Student(0," ");

s[i].accept();

}

System.out.println("the details of student is ");

for(int i=0;i<num;i++)

{

s[i].display();

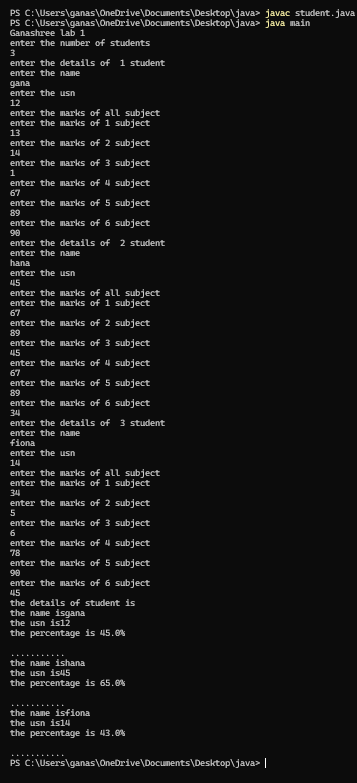
System.out.println("\n........... ");

}

}

}

OUTPUT:



1. Create a class Book that contains four members: name, author, price, and num\_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects

CODE:

import java.util.Scanner;

class Book

{

String name;

String author;

int price;

int num\_pages;

Book()

{}

Book(String name,String author,int price,int num\_pages)

{

this.name=name;

this.author=author;

this.price=price;

this.num\_pages=num\_pages;

}

void set()

{

Scanner input=new Scanner(System.in);

System.out.println("enter the name of the book");

this.name=input.nextLine();

System.out.println("enter the name of author");

this.author=input.nextLine();

System.out.println("enter the number of pages");

this.num\_pages=input.nextInt();

System.out.println("enter the price of book");

this.price=input.nextInt();

}

public String toString()

{

String name,author,price,num\_pages;

name="Book name is "+this.name+"\n";

author="author name is "+this.author+"\n";

price="price is "+this.price+"\n";

num\_pages="number of pages is "+this.num\_pages+"\n";

return name+author+price+num\_pages;

}

}

class main

{

public static void main(String args[])

{

Scanner input=new Scanner(System.in);

System.out.println("enter the number of books");

int num=input.nextInt();

Book[] b=new Book[num];

for(int i=0;i<num;i++)

{

b[i]=new Book();

b[i].set();

}

for (int i=0;i<num;i++)

{

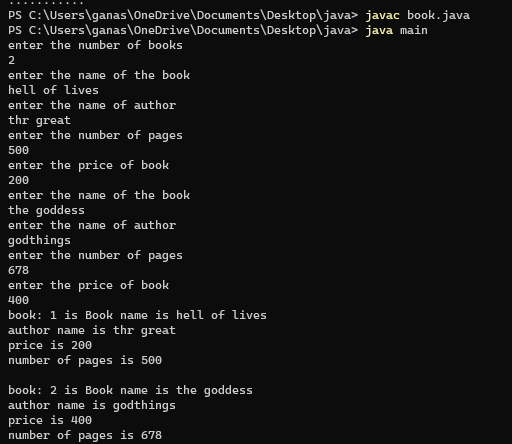
System.out.println("book: "+(i+1)+" is "+b[i]);

}

}

}

Output:



1. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain the method printArea( ) that prints the area of the given shape.

CODE:

abstract class Shape {

protected int value1;

protected int value2;

public Shape(int value1, int value2) {

this.value1 = value1;

this.value2 = value2;

}

public abstract void printArea();

}

class Rectangle extends Shape {

public Rectangle(int length, int width) {

super(length, width);

}

public void printArea() {

int area = value1 \* value2;

System.out.println("Area of Rectangle is: " + area);

}

}

class Triangle extends Shape {

public Triangle(int base, int height) {

super(base, height);

}

public void printArea() {

double area = 0.5 \* value1 \* value2;

System.out.println("Area of Triangle is: " + area);

}

}

class Circle extends Shape {

public Circle(int radius) {

super(radius, 0);

}

public void printArea() {

double area = Math.PI \* value1 \* value1;

System.out.println("Area of Circle is: "+ area);

}

}

class Main {

public static void main(String[] args) {

Rectangle rectangle = new Rectangle(5, 7);

rectangle.printArea();

Triangle triangle = new Triangle(5, 8);

triangle.printArea();

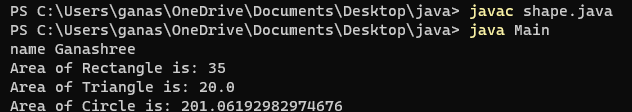
Circle circle = new Circle(8);

circle.printArea();

}

}

Output:



1. Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. 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, impose penalty if necessary and update the balance.

CODE:

class Bank {

public static void main(String[] args) {

SavingsAccount savingsAccount = new SavingsAccount("diksha n", "SA1001");

CurrentAccount currentAccount = new CurrentAccount("hanwitha b", "CA2002");

savingsAccount.deposit(5000);

savingsAccount.displayBalance();

savingsAccount.computeInterest();

savingsAccount.displayBalance();

savingsAccount.withdraw(2000);

savingsAccount.displayBalance();

currentAccount.deposit(8000);

currentAccount.displayBalance();

currentAccount.withdraw(5000);

currentAccount.displayBalance();

}

}

class Account {

protected String customerName;

protected String accountNumber;

protected double balance;

public Account(String customerName, String accountNumber) {

this.customerName = customerName;

this.accountNumber = accountNumber;

this.balance = 0;

}

public void deposit(double amount) {

balance += amount;

System.out.println("Deposit of $" + amount + " successful");

}

public void displayBalance() {

System.out.println("Account Number: " + accountNumber + "\nBalance: $" + balance);

}

}

class SavingsAccount extends Account {

public SavingsAccount(String customerName, String accountNumber) {

super(customerName, accountNumber);

}

public void computeInterest() {

double interestRate = 0.05;

double interest = balance \* interestRate;

balance += interest;

System.out.println("Interest of $" + interest + " computed and added to the balance");

}

public void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Withdrawal of $" + amount + " successful.");

} else {

System.out.println("Insufficient funds for withdrawal.");

}

}

}

class CurrentAccount extends Account {

private double minimumBalance = 1000;

public CurrentAccount(String customerName, String accountNumber) {

super(customerName, accountNumber);

}

public void withdraw(double amount) {

if (balance - amount >= minimumBalance) {

balance -= amount;

System.out.println("Withdrawal of $" + amount + " successful.");

} else {

System.out.println("Insufficient funds. Service charge applied.");

imposeServiceCharge();

}

}

private void imposeServiceCharge() {

double serviceCharge = 20;

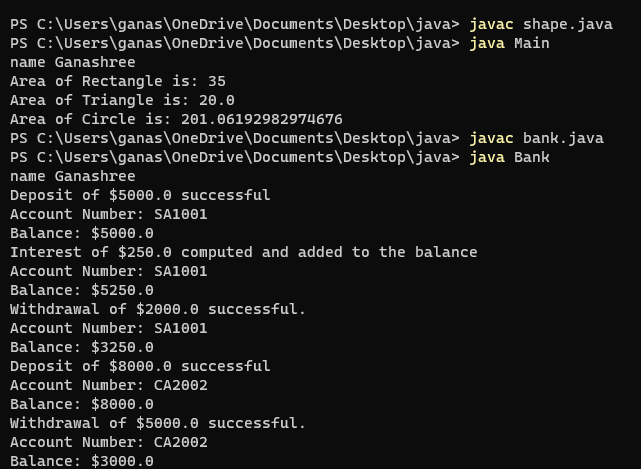
balance -= serviceCharge;

System.out.println("Service charge of $" + serviceCharge + " imposed.");

}

}

OUTPUT:



1. Package program

Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class internals derived from student has an array that stores the internal marks scored in five 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 five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students.

CODE:

package cie;

public class Student{

public String name;

public String usn;

public int sem;

}

package cie;

public class Internals extends Student{

public int[] marks=new int[5];

}

package see;

import cie.Student;

public class External extends Student{

public int[] seemarks=new int[5];

}

import cie.Internals;

import see.External;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

System.out.println("Enter the number of students");

Scanner input = new Scanner(System.in);

int n = input.nextInt();

Internals[] s1 = new Internals[n];

External[] s2 = new External[n];

int[] finalcie = new int[n];

int[] finalsee = new int[n];

for (int i = 0; i < n; i++) {

s1[i] = new Internals();

System.out.println("Enter the name");

s1[i].name = input.next();

System.out.println("Enter the usn");

s1[i].usn = input.next();

System.out.println("Enter the sem");

s1[i].sem = input.nextInt();

System.out.println("Enter the marks of 5 subjects");

for (int j = 0; j < 5; j++) {

s1[i].marks[j] = input.nextInt();

finalcie[i] += s1[i].marks[j];

}

}

for (int i = 0; i < n; i++) {

s2[i] = new External();

System.out.println("Enter the name");

s2[i].name = input.next();

System.out.println("Enter the usn");

s2[i].usn = input.next();

System.out.println("Enter the sem");

s2[i].sem = input.nextInt();

System.out.println("Enter the marks of 5 subjects");

for (int j = 0; j < 5; j++) {

s2[i].seemarks[j] = input.nextInt();

finalsee[i] += s2[i].seemarks[j];

}

}

System.out.println("Final marks:");

for (int i = 0; i < n; i++) {

System.out.println("Name: " + s1[i].name + " USN: " + s1[i].usn + " Sem: " + s1[i].sem);

System.out.println("Internal marks: " + finalcie[i]);

System.out.println("External marks: " + finalsee[i]);

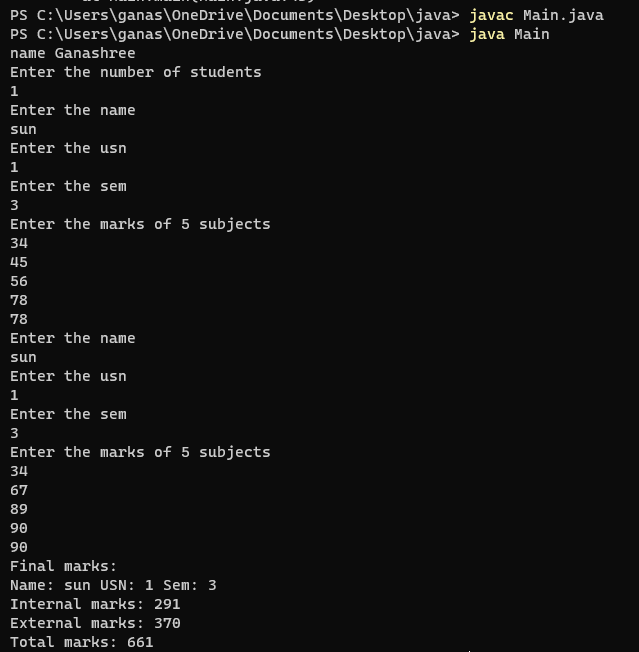
System.out.println("Total marks: " + (finalcie[i] + finalsee[i]));

}

}

}

OUTPUT:



1. Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

CODE:

class WrongAgeException extends Exception {

public WrongAgeException(String message) {

super(message);

}

}

class Father {

private int age;

public Father(int age) throws WrongAgeException {

validateAge(age);

this.age = age;

}

private void validateAge(int age) throws WrongAgeException {

if (age < 0) {

throw new WrongAgeException("Father's age cannot be negative");

}

}

public int getAge() {

return age;

}

}

class Son extends Father {

private int sonAge;

public Son(int fatherAge, int sonAge) throws WrongAgeException {

super(fatherAge);

validateSonAge(sonAge);

this.sonAge = sonAge;

}

private void validateSonAge(int sonAge) throws WrongAgeException {

if (sonAge >= getAge()) {

throw new WrongAgeException("Son's age should be less than Father's age");

}

}

public int getSonAge() {

return sonAge;

}

}

class exceptionrun {

public static void main(String[] args) {

try {

Father father = new Father(-5);

} catch (WrongAgeException e) {

System.out.println("Exception: " + e.getMessage());

}

try {

Son son = new Son(30, 35);

} catch (WrongAgeException e)

{

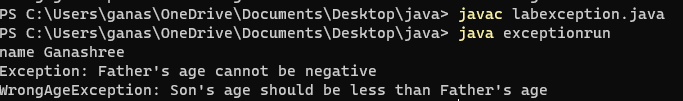
System.out.println(e);

}

}

}

Output:



1. Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

Code:

class BMSCE extends Thread {

public void run() {

while (true) {

System.out.println("BMS COllege OF ENgineering");

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

}

class CSE extends Thread{

public void run() {

while(true){

System.out.println("CSE");

try

{

Thread.sleep(2000);

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

}

class run{

public static void main(String[] args){

BMSCE b=new BMSCE();

CSE c=new CSE();

b.start();

c.start();

}

}

Output:

