

OBSERVATION COPY:

Program 1:

PROGRAM-1

Develop a Java program that prints all roots according to the quadratic equation $ax^2+bx+c=0$ here a, b, c are user input. Use the quadratic formula if the discriminant b^2-4ac is ≥ 0 , display a message stating that there are no real solutions.

```

import java.util.Scanner;

class coeff {
    double a;
    double b;
    double c;
}

public class Quadratic {
    public static void main (String[] args) {
        Scanner scanner = new Scanner(System.in);
        coeff coeff = new coeff();

        System.out.println("Enter coeff. of a, b, c");
        System.out.print("Enter a:");
        coeff.a = scanner.nextDouble();
        while (coeff.a == 0) {
            System.out.println("Not a quad. eqn, enter non-zero value");
            coeff.a = scanner.nextDouble();
        }
        System.out.print("Enter b:");
        coeff.b = scanner.nextDouble();
    }
}

```

```

        System.out.print("Enter c:");
        coeff.c = scanner.nextDouble();

        double d = coeff.b * coeff.b - 4 * coeff.a * coeff.c;

        if (d == 0) {
            double r1 = -coeff.b / (2 * coeff.a);
            System.out.println("Roots are real & equal");
            System.out.println("Root 1 & Root 2: " + r1);
        } else if (d > 0) {
            double r1 = (-coeff.b + Math.sqrt(d)) / (2 * coeff.a);
            double r2 = (-coeff.b - Math.sqrt(d)) / (2 * coeff.a);
            System.out.println("Roots are real & distinct");
            System.out.println("Root 1: " + r1);
            System.out.println("Root 2: " + r2);
        } else {
            double realpart = -coeff.b / (2 * coeff.a);
            double imaginarypart = Math.sqrt(-d) / (2 * coeff.a);
            System.out.println("Roots are imaginary");
            System.out.println("Root 1: " + realpart + " + i" + imaginarypart);
            System.out.println("Root 2: " + realpart - " + i" + imaginarypart);
        }
    }
}

```

Output :-

```

Enter coefficient a, b, c:
Enter coefficient a: 8
Enter coefficient b: 50
Enter coefficient c: 20

Roots are real and distinct
Root 1: -0.1295
Root 2: -5.87048

Enter coefficient a, b, c:
Enter a: 3
Enter b: 4
Enter c: 5
Roots are imaginary
Root 1: -0.66 + i.105541
Root 2: -0.66 - i.105541

Enter coefficient a, b, c:
Enter a: 1
Enter b: 2
Enter c: 5
Roots are imaginary.

```

21/11/21

Program 2:

PROGRAM-2.

Develop a Java program to create a class Student with members `un`, `name`, an array `crs` and an array `marks`. Include methods to accept & display details and a method to calculate $\% \text{sgpa}$ of a student.

```
import java.util.Scanner;
```

```
class subject {
    int subjectmarks;
    int credits;
    int grade;
}
```

```

class student {
private static final int no_of_subjects = 5;
    String name;
    String uen;
    double GPA;
    Subject[] subjects;
    Scanner sc;
}

```

```
Student (graph scanner) {
    subjects = new Subject [no-ef-subjects];
    for (int i=0; i<no-ef-subjects; i++) {
        subjects[i] = new Subject(i);
    }
}
```

this. sc = scanner;

```
void getStudentDetail () {
    system.out.println ("enter name & USN:");
    name = sc.nextLine();
    usn = sc.nextLine();
}
```

double gas marks (2) $\frac{1}{2}$
system. each pinion (1" each mark & credit of
each subject "2";
double total credits = 0;
double total grade points = 0;

```

yes (int i=0; i < no_of_subjects; i++) {
    double mark = sc.getMark(i);
    double credits = sc.getCredits(i);
}

```

```
if (marks < 0 || marks > 100) {
    system_exit("invalid marks");
}
//
continue;
```

```
if (marks == 100) {
    subjects[6].subjectMarks = 10;
```

```
else if (marks >= 90) {
    subject [C]. subjectmarks = 10;
```

else if (marks \geq 80) &
subject 16]. Subject marks = 9;

else if (marks >= 70) {
subjects[17].sum = marks;

else if (marks \geq 60) {
 Subjects [i]. Subject marks = 7;

```

else if (marks >= 50) {
    subject[i].subjectmarks = 6;
}

```

```
    } else if (marks >= 40) {  
        subject[i].subjectmarks = 5;
```

case of

Subjects [i]. Subject Marks = 0.

Subjects [17]. credits = (ins) credit;

```

subjects[i].grade = subjects[i].credits *
    subjects[i].submarks;
totalcredits += subjects[i].credits;
totalgradeperc += subjects[i].grade;

```

```
SGPA = totalGradePoints / totalCredits;
return totalCredits;
```

```
void computeSQPA() {
    System.out.println("SQPA:" + SQPA);
}
```

```

// Student class
public class Student {
    private String name;
    private int rollNo;
    private float avg;

    // constructor
    Student() {
        name = " ";
        rollNo = 0;
        avg = 0.0f;
    }

    // setter methods
    public void setName(String name) {
        this.name = name;
    }

    public void setRollNo(int rollNo) {
        this.rollNo = rollNo;
    }

    // getter methods
    public String getName() {
        return name;
    }

    public int getRollNo() {
        return rollNo;
    }

    // calculate average
    public void calculateAvg() {
        // ...
    }
}

```

Output:

Enter number of students : 2

Processing details for student 1:

Enter name and roll no :

SAM
18M23CS295

Enter marks and credits of each subject :

95	3
87	4
99	2
81	1
93	3

SQPA : 9.615

Processing details for student 2:

Enter name and roll no :

Ram
18M23CS300

Enter marks and credits of each subject :

98	3
78	3
89	4
90	1
70	2

SQPA : 8.9231

21/11/24

Program 3:

PROGRAM-3

create a class book which contains four members: name, author, price, num-pages. provide 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 main program to create n book objects.

```
import java.util.Scanner;

class book {
    String name;
    String author;
    int price;
    int numpages;

    book(String name, String author, int price,
          int numpages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numpages = numpages;
    }

    public String toString() {
        return "Book name: " + this.name + "\n" +
            "Author name: " + this.author + "\n" +
            "Price: " + this.price + "\n" +
            "Number of pages: " + this.numpages + "\n";
    }
}
```

```
public class main {
    public static void main (String[] args) {
        Scanner s = new Scanner(System.in);
        int n;
        System.out.println("Enter no. of books:");
        n = s.nextInt();
        s.nextLine();

        book[] b = new book[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Book " + (i+1) + ":");
            System.out.print("Enter name of book:");
            String name = s.nextLine();
            System.out.print("Enter author of book:");
            String author = s.nextLine();
            System.out.print("Enter price of book:");
            int price = s.nextInt();
            System.out.print("Enter no. of pages in book:");
            int numpages = s.nextInt();
            s.nextLine();

            b[i] = new book(name, author, price,
                           numpages);
        }

        System.out.println("\n Book details:");
        for (book book : b) {
            System.out.println(book.toString());
        }
        s.close();
    }
}
```

Output:

Enter number of books :
2

Book 1:

enter name of book: ~~the~~ The Alchemist
enter author of book: ~~Paulo~~ Paulo
enter price of book: 250
enter number of pages in book: 200

Book 2:

enter name of book: The Grapes of Wrath
enter Author of book: John
enter price of book: 300
enter number of pages: 400

Book details

Book name: The Alchemist
Author name: Paulo
Price: 250
no. of pages: 200

Book name: The Grapes of Wrath
Author name: John
Price: 300
No. of Pages: 400

Program 4:

```

PROGRAM-4

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

import java.util.Scanner;

abstract class shape {
    double a;
    double b;
    double result;

    abstract void printArea();
}

class Rectangle extends shape {
    void printArea() {
        System.out.println("Enter dimension of
        rectangle");
        Scanner s = new Scanner(System.in);
        a = s.nextDouble();
        b = s.nextDouble();
        result = a * b;
        System.out.println("Area: " + result + "sq units");
    }
}

```

```

}

class Triangle extends shape {
    void printArea() {
        System.out.println("Enter dimension of
        triangle");
        Scanner s = new Scanner(System.in);
        a = s.nextDouble();
        b = s.nextDouble();
        result = (a * b) / 2;
        System.out.println("Area: " + result +
        "sq units");
    }
}

class Circle extends shape {
    void printArea() {
        System.out.println("Enter radius of circle");
        Scanner s = new Scanner(System.in);
        a = s.nextDouble();
        result = 3.142 * a * a;
        System.out.println("Area: " + result + "sq units");
    }
}

class PrintArea {
    public static void main (String args[]) {
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();
        r.printArea();
        t.printArea();
        c.printArea();
    }
}

```

Output:

Enter dimension of rectangle

5 20

Area: 100.0 sq units

Enter dimension of triangle

5 10

Area: 25.0 sq units

Enter radius of circle

22

Area: 1520.7279 sq units

21/11/24

Program 5:

```
import java.util.Scanner;

class Account {
    String CustomerName;
    int AccountNumber;
    double balance;

    Account(String name, int accno) {
        CustomerName = name;
        AccountNumber = accno;
        balance = 0.0;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount +
            "\n Updated balance: " + balance);
    }

    void displayBalance() {
        System.out.println("Account balance: " +
            balance);
    }

    void withdraw(double amount) {
        System.out.println("Withdrawal is specific
            to account type");
    }

    void computeInterest() {
        System.out.println("Interest can't be
            calculated for this account type");
    }
}
```

```
class SavingsAccount extends Account {
    double InterestRate = 0.04;

    SavingsAccount(String name, int accno) {
        super(name, accno);
    }

    void computeInterest() {
        double Interest = balance * InterestRate;
        balance += Interest;
        System.out.println("Interest added: " +
            Interest + "\n Updated balance: " + balance);
    }

    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal: " +
                amount + "\n Updated balance: " + balance);
        }
        else {
            System.out.println("Insufficient balance");
        }
    }
}

class CurrentAccount extends Account {
    double minimumBalance = 500.0;
    double serviceCharge = 50.0;

    CurrentAccount(String name, int accno) {
        super(name, accno);
    }
}
```

```
super(name, accno);

void checkMinimumBalance() {
    if (balance < minimumBalance) {
        System.out.println("Balance is
            below minimum. Service charge imposed");
        balance -= serviceCharge;
        System.out.println("Service charge: " +
            serviceCharge + "\n Updated balance: " +
            balance);
    }
}

void withdraw(double amount) {
    if (balance >= amount) {
        balance -= amount;
        System.out.println("Withdrawal: " +
            amount + "\n Updated balance: " +
            balance);
        checkMinimumBalance();
    }
    else {
        System.out.println("Insufficient
            balance");
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter customer name:");
        String name = sc.nextLine();
        System.out.print("Enter acc no:");
    }
}
```



```

int acno = co.nextline();
co.nextLine();
System.out.print("Enter account type  
(savings/current): ");
String accType = co.nextline();

Account account;
if (accType.equalsIgnoreCase("savings")) {
    account = new SavingsAccount(name, acno);
} else if (accType.equalsIgnoreCase("current")) {
    account = new CurrentAccount(name, acno);
} else {
    System.out.println("Invalid");
    co.close();
    return;
}

while (true) {
    System.out.println("1. Menu");
    System.out.println("2. Deposit");
    System.out.println("3. Withdraw");
    System.out.println("4. Display Balance");
    System.out.println("5. Display Interest");
    System.out.println("6. Exit");
    System.out.print("Enter your choice: ");
    int choice = co.nextInt();
}

```

```

switch (choice) {
    case 1:
        System.out.print("Enter deposit amount: ");
        double depositAmount = co.nextDouble();
        account.deposit(depositAmount);
        break;

    case 2:
        System.out.print("Enter withdrawal amount: ");
        double withdrawAmount = co.nextDouble();
        account.withdraw(withdrawAmount);
        break;

    case 3:
        account.displayBalance();
        break;

    case 4:
        account.displayInterest();
        break;

    case 5:
        System.out.println("Exiting");
        co.close();
        return;

    default:
        System.out.println("Invalid choice");
}
}
}
}

```

Output :-

Enter customer name: sam
Enter account number: 123
Enter account type (savings/current): savings

Menu:

1. Deposit
2. Withdraw
3. Display Balance
4. Display Interest
5. Exit

Enter your choice: 1
Enter deposit amount: 500
Deposited: 500.0
Updated Balance: 500.0

Enter your choice: 4
Interest added: 20.0
Updated Balance: 520.0

Enter your choice: 2
Enter withdrawal amount: 10
Withdrawn: 10.0
Updated Balance: 510.0

Enter customer name: xyz
Enter account number: 134
Enter account type (savings/current): current

Enter your choice: 1
Enter deposit amount: 210
Deposited: 210.0
Updated balance: 210.0

Enter your choice: 2
Enter withdrawal amount: 10
Withdrawn: 10.0
Updated Balance: 200.0
Balance is below minimum. Service Charge imposed.
Service Charge: 50.0
Updated balance: 150.0

21/11/21

Program 6:

PROGRAM 6:

Create a package C16 which has two classes - Student and Internals. The class Student has members like user, name, sem. The class Internals has an array that stores the internal marks scored in the classes of the current semester of the student. Create another package SEC which has the class ExternalMarks which is a derived class of Student. This class has an array that stores the SEC marks scored in the classes of current semester of the student. Import the two packages in a file that declares the global marks of a student in all five classes.

```
package C16;
public class Student {
    String user;
    String name;
    int sem;

    public Student (String user, String name,
        int sem) {
        this.user = user;
        this.name = name;
        this.sem = sem;
    }
}
```

```
package C16;

public class Internals {
    int[] internalMarks = new int[5];

    public Internals (int[] marks) {
        if (marks.length == 5) {
            for (int i=0; i<5; i++) {
                internalMarks[i] = marks[i];
            }
        } else {
            System.out.println ("provide exactly
                5 marks for Internals");
        }
    }

    void displayInternalMarks () {
        System.out.print ("Internal Marks:");
        for (int i=0; i< internalMarks.length;
            i++) {
            System.out.print (internalMarks[i] +
                " ");
        }
        System.out.println ();
    }
}
```

```
package SEC;

import C16.Student;
public class ExternalMarks extends Student {
    int[] externalMarks = new int[5];

    public External (String user, String name,
        int sem, int[] marks) {
        super (user, name, sem);
        if (marks.length == 5) {
            for (int i=0; i<5; i++) {
                externalMarks[i] = marks[i];
            }
        } else {
            System.out.println ("provide exactly
                5 marks for SEC");
        }
    }

    void displayExternalMarks () {
        System.out.print ("SEC Marks:");
        for (int i=0; i< externalMarks.length;
            i++) {
            System.out.print (externalMarks[i] +
                " ");
        }
        System.out.println ();
    }
}
```

```
import C16.Student;
import C16.Internals;
import SEC.External;

import java.util.Scanner;

public class Main {
    public static void main (String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print ("enter name of student:");
        int n = scanner.nextInt();

        Student[] students = new Student[n];
        Internals[] internals = new Internals[n];
        External[] externals = new External[n];

        for (int i=0; i<n; i++) {
            System.out.print ("enter details for student " +
                (i+1) + ": ");
            String user = scanner.next();
            String name = scanner.next();
            String sem = scanner.next();
            int sem = scanner.nextInt();

            students[i] = new Student (user, name,
                sem);
        }
    }
}
```



```

s.o.pln ("Enter internal marks for 5 subjects");
int i; internalMarks = new int[5];
for (int j=0; j<5; j++) {
    externalMarks[j] = scanner.nextInt();
}
internals[i] = new External (user.name,
                             sem, externalMarks);
}

System.out.println ("Final marks of
                      students:");
for (int i=0; i<n; i++) {
    students[i].display();
    internals[i].displayInternalMarks();
    externals[i].displayExternalMarks();

    s.o.p ("Final marks:");
    for (int j=0; j<5; j++) {
        int finalMark = internals[i].
                        internalMarks[j] +
                        externals[i].externalMarks[j];
        s.o.p (finalMark + " ");
    }
}

scanner.close();
}
}

```

Output:-

Enter number of students : 1

Enter details:

UIN : 16M23

Name : Sam

Semester : 3

Enter internal marks for 5 subjects:

80 37 38 29 28

Enter sec marks for 5 subjects:

99 98 87 86 90

Final Marks of students:

UIN : 16M23

Name : Sam

Semester : 3

Internal Marks : 80 37 38 29 28

Sec Marks : 99 98 87 86 90

21/4/24

Program 7:

PROGRAM 7

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 age and throws the exception "WrongAge" when the input age < 0. In son class, implement a constructor that uses both father and son's age and throws an exception if son's age is >= father's age.

```

import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException (int age) {
        super ("Age cannot be negative. Invalid age: " + age);
    }
}

class InvalidSonAgeException extends Exception {
    public InvalidSonAgeException (int fatherAge, int sonAge) {
        super ("Son's age can't be greater than or equal to father's age.");
    }
}

```

```

class Father {
    int fatherAge;

    public Father (int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException (age);
        }
        this.fatherAge = age;
    }

    class son extends Father {
        int sonAge;

        public son (int fatherAge, int sonAge) throws WrongAgeException, InvalidSonAgeException {
            super (fatherAge);
            if (sonAge < 0) {
                throw new WrongAgeException (sonAge);
            }
            if (sonAge >= fatherAge) {
                throw new InvalidSonAgeException (fatherAge, sonAge);
            }
            this.sonAge = sonAge;
        }
    }
}

```

```

public void displayAge () {
    System.out.println ("Enter father's age: " + fatherAge);
    System.out.println ("Enter son's age: " + sonAge);
}

public class ExceptionHandlingInheritance {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);

        try {
            System.out.println ("Enter father's age: ");
            int fatherAge = sc.nextInt ();

            System.out.println ("Enter son's age: ");
            int sonAge = sc.nextInt ();

            son son = new son (fatherAge, sonAge);
            son.displayAge ();
        } catch (WrongAgeException | InvalidSonAgeException e) {
            Scanner.close ();
        }
    }
}

```

Output :-

Enter father's Age : 23
Enter son's Age : 25

Son's age cannot be greater than or equal to father's age.

Enter father's Age : 40
Enter son's Age : 12
father's age : 40
son's age : 12

21/11/21

Program 8:

```

class DisplayThread extends Thread {
    String message;
    int interval;

    public DisplayThread (String message, int interval) {
        this.message = message;
        this.interval = interval;
    }

    public void run () {
        try {
            while (true) {

```

```

    s.o. pin (message);
    Thread.sleep (interval + 1000); } 3
    catch (InterruptedException e)

    { s.o. pin ("Thread interrupted : "+ message); }

    public static void main (String[] args) {
        DisplayThread thread1 = new DisplayThread
            ("AMSC", 10);
        DisplayThread thread2 = new DisplayThread
            ("CSC", 2);
    }

```

Output :-

BMS College of engineering
CSE
CSE
CSE
CSE
CSE

BMS College of engineering
CSE
CSE
CSE
CSE
CSE

BMS College of engineering