1) Develop a java program that prints all real solution to the quadratic equations $ax^2+bx+c=0$ read in a,b,c and use the quadratic formula

Bafna Gold
Date: 1/1/8 Page:

$$d = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$$

```
import Java.util.Scanner;
class first {
public static void main (String args[]) {
    int a,b,c;
    double r1,r2;
    system.out.println("Enter the quadratic equation");
    Scanner s1 = new scanner();
    a=s1.nextline();
    b = s1.nextline();
    c = s1.nextline();
    if (a==0) {
        system.out.println(" cofficient are invalid ");}
    else {
        d=b^2-(4*a*c)
        if (d>0) {
d=1     system.out.println("It is real and distinct roots");
            r1 ((-b)+ Math.sqrt(d)))/(2*a)
            r2=((-b + Math.sqrt(d))/(2*a);
            system.out.println("The roots are"+r1+"and"+r2)
        }
        else if (d==0) {
            system.out.println("It has real and equal roots")
            r1=(-b)/2*a;
            system.out.println("The roots "+r1+"and"+r1);
        else {
            system.out.println("It was no real roots");
        }
    }
}
}
```

$d = b^2 - 4ac$
$1 - 4 \times 1 \times 1$
$1 - 4$
$= -3$

$\frac{-b - 4ac}{2a}$

Output:- Enter the cofficients of quadratic equation
1
2
1

it has a real and equal roots

the roots are -1.0 and -1.0

Enter the coefficients of quadratic equation

1

1

4

It has no real roots

Enter the coefficients of quadratic equation

1

4

1

It is real & distinct roots

the roots are -0.2679 and -3.730508

Bafna Gold
Date: / / Page:

8) Develop Java program to create student with members usn, name, an array.
credit & an array marks, include methods to accept & display details and a method to calculate sgpA of a student.

```
-> import java.util.*;
class student {
    string USN;
    String Name;
    int credits[]
    int Marks[]

    void accept() {
        Scanner s1 = new Scanner(System.in);
        system.out.println("Enter the USN:=");
            USN = s1.next~~Int~~();
        system.out.println("Enter the name:=");
        NAME = s1.next();
        system.out.println("Enter the no of subject:=");
        int s3 = s1.nextInt();

        credits = new int[s3];
        Marks = new int[s3];
        s1.nextLine();

        for (int i=0, i<s3; i++) {
            system.out.println("Enter the credits:"+(i+1)+":");
            credits[i] = s1.nextInt();
            System.out.println("Enter the subject"+(i+1)+":");
            Marks[i] = s1.nextInt();
        }
    }
```

```
void display(){
        s.o.pln ("usn"+usn);
        s.o.pln ("NAME"+NAME);
        for (int i=0; i<credits.length; i++){
            system.out.println ("credits"+(i+1)+":"+credits[i])
        }
    }
double calcgpa(){
        double sgpa=0;
        for(int i=0; i<credits.length; i++){
            sgpa = credit[i] * Marks[i]; }
        sgpa /= credits * Marks[i];
            return sgpa;
        }
    }

class std{
    P S V M (string args[]){
        student s=new student();
        s.accept();
        s.display();
        system.out.println ("SGPA:"+s.calcgpa());
```

Enter the usn:-
2023BMS02551

Enter the ~~credits for su~~ NAME:
Pooja

Enter the credits for subject 1:
4

Enter the Marks subject 1
40

Enter the credits for subject 1:
7

Enter the Marks subject 2:
26

USN:- 2023BMS
NAME:- Pooja
credits :1- 4
credits :2:- 7
SGPA = 56

8/4/2024

3) Create a class Book which contains 4 members, author, price, numpage include a constructor to set the values for members. include methods to set and get the details of the object. include a to string() method that could display the complete details of the books. Develop a java program to create n book objects

```
-> Public class Book{
Private
~~public~~ string name;
private string another;
private double price;
  private int numpages;

public class (String name, string another, double price, int
 numpages) {
   This.name = name;
   this.another = another;
   This.price = price;
   This numpages = numpages; }

public string tostring() {
      return "Book details: \n name: " + name + "\n Another:"
        + Author + "\n Price: " + Price + "\n no of pages" +
          numpages);
      }
public static void main (String args[]){
    int n=3;
    Book[] books = new books[n]

    for (int i=0; i<n; i++) {
        book[i] = new books("Book " + (i+1), " Another " +(i+1),
          200+i*5, 100+i*50);
    }
```

```
for (int i=0 ; i<n; i++){
        System.out.println (book[i].toString());
        System.out.println ("------");
    }
}
}
```

Output →

* Book details →
name :- book 1
Author :- Author 1
price :- 20.0
no of pages :- 1000

* Book details :
name : book 2
Author :- Author 2
price :- 20.0
no of pages - - 150

* books details :-
name = book 3
Author = Author 3
price = 30.0
no of ages = 20.0

4) Develop a Java program to create an abstract class named shape that contains 2 integers & an empty method named printArea(). provide 3 classes named rectangle, Triangle & circle such that each one of the classes Extends the class shape. each one

```
abstract class shape {
    protected int dimension1;
    protected int dimension2;

Public shape (int dimension1, int dimension2) {
        This. dimension1 = dimesion2;
        this. dimension2 = dimension2;
    }.
public abstract void printArea();
}
class rectangle extends shape {
    public rectangle (int length, int width) {
            super (length, width);
    }
public void printArea() {
    int area = dimension1 * dimension2;
    System.out. Println ("rectangle Area:" + area);
    }
}

class Tringle extends shape {
    print Triangle (int base, int height) {
        super (base, height);
    }
public void print Area() {
    double area = 0.5 * dimension1 * dimension2;
    S.O.pln ("Tringle Area " + area);
}
}
```

```
class circle extends shape {
    public circle (int radius){
        super(radius,0);
}
public class Main {
        P S V M(string args[]){
        Rectangle R = new Rectangle(5,8);
        Triangle T= new Triangle(2,9);
        circle c = new circle(2);
        R.printArea();
        T.printArea();
        c.printArea();
}
}
```

~~Part-a~~ O/P - 9

Output →

Rectangle Area : 50

Triangle Area : 90

circular Area : 50.2654

22/1/2024

5) Create package CIE which has two classes - student and Internals. The class personal has members like USN, name, SEM. The class internals 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 Package SEE which has an array that stores of the current semester of the student. Import the 2 package in the file that declares the final marks of n students in all five courses.

```
Package CIE
Public Class student {
    protected string USN;
    protected ~~name~~ string NAME;
    protected int SEM;

public student ( string USN, string NAME, int SEM) {
        This.USN= USN;
        This.NAME = NAME;
        This.SEM = SEM; }
public string getUSN() {
        return USN }
public ~~str~~ void setUSN (string USN) {
        This.USN = USN; }
public string getNAME() {
        return NAME; }
public void setName( string NAME) {
        This.NAME = NAME; }
public ~~string~~ int getSEM () {
        return SEM; }
public void setSEM (int SEM) {
        This.SEM = SEM;
    }
}
Package CIE;
Public class Internals {
    Protected int[] internalMarks = new int[5];
```

```
public Internals (int[] internalMarks )
        This. internalMarks = internalMarks }

public int[] getInternal Marks (){
        return internalMarks;
  }
public void setInternalMarks (int[] internalMarks){
        This. internalMarks = internalMarks;
  }
}
```

(2)

```
package SEE;
  import CIE. student;
  public class External ~~Extends~~ student {
        protected int [] seeMarks = new int [5];
  public External (string usn, string name, int sem, int[] seeMarks
    super (usn, name, sem);
      This. seeMarks = seeMarks;
  }
  public int[] get Marks (){
        return seeMarks; }
  public void setSeeMarks ( int[] seeMarks ){
        This. seeMarks = seeMarks;
  }
}
import CIE. student;
import CIE. Internals;
import SEE. External;

public class Main file {
      p s v M (string args[]){
        int n=5;
        External [] students = new External[n];
```

```
for (int i=0 ; i<n; i++){
        int [] internalMarks = {50+i, 52+i, 55+i, 48+i, 60+i};
        int [] seeMarks={80+i, 65+i, 70+i, 55+i, 80+i}
  student [i] = new External ("USN" +i, "student" +i, 3,
        seeMarks);
}
for (int i=0 ; i<n; i++){
        s.o.pln ("student" + students [i].getName());
        s.o.pln ("USN: " + students [i]. getUSN());
        s.o.pln ("SEM" + students [i]. getSem());

  s.o.pln ("Internal Marks:");
     for (int j=0, j<5; j++){
            s.o.pln ("course " + (j+1) + ":" + students [i].
            getInternal Marks () [j]);
        }
  s.o.pln ("SEE Marks");
        for (int j=0; j<5 ; j++){
            s.o.pln ("course" + (j+1) + ":" + students [i].
            getSeeMarks () [j]);
     }
     s.o.pln ();
  }
 }
}
```

O/P -> Enter the no of student :- 1

~~student :-~~ Details for student 1:

USN : USN001

NAME : ABC

SEM : 3

Internal Marks:

course 1 : 75

course : 80

course 3 : 75

course 4 : 80

Courses : 90

External Marks

course 1 : 80

course 2 : 85

Course 3 : 80

course 4 : 85

course 5 : 90

Total Marks : 820

Percentage : 82%

cgpa : 8.2

22/1/2024

7) write a program that demonstrats handing of exceptions in inheritance tree create base class called father and derived class called son which extends the base class. In father class implement a constructor which take the age & throws the exception wrongAge() when the I/P age<0. In cases both father & son age and throws an exception if sons age is >= fathers age.

```
import java.util.scanner;
class wrongException extends Exception {
    public wrongAgeexception (string Message){
        super (Message);
    }
}

class father {
    private int age;
    public father (int age) throws WrongException {
     if (age<0) {
        Throw new WrongAgeException ("Age is can't be
        negative");
    }
     This.age = age;

public int getAge(){
        return Age;
    }
}

class son extends father {
    private int SonAge;
    public son (int fatherAge, int SonAge) throws wrongAgeEXCEP
                                                  tion {
     super ( father Age);
     if (sonAge >= fatherAge){
        Throw new wrongAgeException (" son's age should be less than
                                        Father age");
    }
     This.sonAge = sonAge;
    }
```

```
public int getAge(){
    return sonAge;
  } }
• Scanner s = new Scanner(System.in);
public class PEExample {
    P S V M (String args[]){
      try { ~~printlt("Enter father age");~~ s.nextInt
        int fatherInput = ~~Integer.parseInt(args[0]);~~ s.nextInt
        int sonAgeInput = ~~Integer.parseInt(args[1]);~~

        Father father = new Father(fatherAgeInput);
        Son son = new Son(fatherAgeInput, sonAgeInput);

      s.o.pln (" father age:- "+ father.getAge());
      s.o.pln (" son's Age :- "+ son.getAge());
    } catch (NumberFormatException e) {
        s.o.pln (" Invalid I/P ");
    } catch (WrongAgeExcept e) {
        s.o.pln (" Error " + e.getMessage());
    }

  }
}
```

O/P-

Enter the father Age:-

34

Enter the CHID Age:-

10

father's age:- 34

son's age: 10

Enter the father Age :- 2

Enter the child age:- 34

Error! son's age should not be less than father's age

8) write a program that displays 2 Threads 1 Thread displaying BMS college of Engineering every 10 seconds & another thread displays CSE every 2 seconds

```
class NewThread implements Runnable
Thread t;
  NewThread()
    {
      t = new Thread (this "Nthread 1");
                s.o.pln ("n" + t)
                t.start();
public void run(){
    try {
      for (int n=5; n>0; n--) {
          ~~s.o.pln("~~
            s.o.pln ("BMS");
            Thread.sleep(10);
          }
      }
    catch (InterruptedException ie){
        s.o.pln ("BMSE Intruppted");
      }
    s.o.pln ("BMSEE quitting");
  }
class BMS {
    P S V M (string args[])
      {
        new NewThread();
          try {
            for (int n=5; n>0; n--){
                s.o.pln ("CSE:");
                Thread.sleep(20);
            }
catch (Intrupted Exception ie){
  s.o.pln ("CSE Intrupted");
```

```
}
  S.o.pln ("CSE quitting");
  }

Thread [NThread, S.Main]
CSE:
BMS
CSE
BMS
BMS
CSE
BMS
CSE
CSE quitting
BMSCE quitting
```

19/2

Bafna Gold
Date: 10/2 Page:

5) ~~Bank~~ develop a Java program to create a class Bank that maintains kinds of account for its customer's one called saving accounts othe current account. The saving account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides ~~compound interest and withdrawal facilities but no chequ~~ cheque book facility but no interest. current account holders should also maintain a minimum balance and if the balance fall below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this drive the classes. cur-acct and sav act to make them more specific to their requirments. Include the necessary methods in order to achieve the following task.

a) accept deposit from customer & update the balance
b) display the balance
c) compute and deposit interest
d) permit withdrawal & update the balance.

->
```
import Java.util.scanner;
class Account {
private string customerName;
private int accountNumber;
private string accountType;
protected double balance;

public Account (string name, int acNo, string type, double initial
Balance) {
    customerName = name;
    accountNumber = acNo;
    accountType = type;
    balance = initialBalance;
}

public void deposit (double amount){
    balance += amount;
```

```
system.out.println ("Deposite successfull, Updated balance:"+
balance);
  }
public void display Balance(){
        system.out.println ("Account Balance:"+ Balance);
}
public void computeInterest(){
  }
Public void withdraw (double amount){
        if (balance >= amount){
            balance -= amount;
system.out.println ("withdroual successfull. update balance:"+
balance); }
  else {
  system.out.println ("Insufficient funds."); }
  }
}
class currentAccount extends Account{
private double minimum Balance;
private double serviceCharge;
public currentAccount (String name, int accNo, double initial
Balance){
  super (name, accNo, "current", initial Balance);
   minimumBalance=100;
   service charge=50;
  }
public void withdraw (double amount){
        if (balance - amount >= minimum Balance){
            balance -= amount;
system.out.println ("withdrawal successfull. update Balance:"+
balance);
  } else {
system.out.println ("Insufficient funds. service charge d
```

```
service charge + "applied"),
        balance -= serviceCharge;
    }
}
public void computeInterest (){
        s.o.pln ("current account does not earn interest");
    }
}
class saving Account extends Account{
  private double interestRate;
  public saving Account (string name, int accNo, double initial
Balance){
    super (name, accNo, "savings", intialBalance);
      interestRate=0.05;
    }
public void computeInterest(){
    double interest = balance* interestRate;
        balance += interest;
            s.o.pln ("Interest computed & deposite updated
    balance" + balance);
  }
}
public class Bank{
  public static void main (String args[]){
    Scanner s1 = new scanner (system.in);
    system.out.println ("Enter your name:");
    String name= Scanner.nextLine();
    system.out.println ("Enter your account number");
    int accountNumber= Scanner.nextInt();
    system.out.println ("Enter initial balance for saving
        account:");
    double savingInitialBalance=Scanner.nextDouble();
    Saving Account SavingAccount = new savingAccount (name,
    accountNumber, savingInitialBalance);
```

```
System.out.println("Enter initial balance for current account");
double currentInitialBalance = Scanner.nextDouble();
CurrentAccount currentAccount = new CurrentAccount(name,
accountNumber, currentInitialBalance);
SavingAccount.displayBalance();
CurrentAccount.displayBalance();
S.o.pln("Enter deposite amount for saving account");
double depositeAmount = Scanner.nextDouble();
SavingAccount.deposite(depositeAmount);
S.o.pln("Enter withdrawal amount for current account");
double withdrawalAmount = Scanner.nextDouble();
CurrentAccount.withdraw(withdrawalAmount);
SavingAccount.displayBalance();
CurrentAccount.displayBalance();
Scanner.close();
}
}
```

output :-

```
Enter your NAME: John
Enter your account NUMBER: 123456
Enter initial balance for saving account: 5000
Enter initial balance for current account: 3000
Account Balance: 5000.0
Account Balance: 3000.0
Enter deposite amount for saving account: 2000
Deposite successfull. update balance: 7000.0
Enter withdrawal amount for current account: 4000
withdrawal successfull. update balance: -1050.0
Account Balance: 7000.0
Account Balance: -1050.0
```

Week-9

Write a program that create a user interface to perform integer divisions. The user enters two numbers in the text fields, NUM1 and NUM2. The division of NUM1 or NUM2 were not an integer, the program would throw an Arithmetic Exception display the exception in a message dialog box.

```
import java.awt.*;
import java.awt.event.*;
public class IntegerDivisionAWT extends Frame
  implements ActionListener {
      private Textfield num2 field, num2 field,
        result field;
      private Button divide Button,
public IntegerDivision AWT (){
    setTitle ("Integer Division"),
    setsize(300, 200);
    setlayout (new flowlayout());
    addwindowlistener (new windowAdapter(){
      public void windowclosing (windowEvent e) {
          dispose();
      }
    });

// create components
label numlable = new label ("NUM1:");
  num1field = new Textfield (10);
  label num2label = new label ("NUM2:");
  num2field = new Textlabel ("10");
  label resultlabel = new label ("Result");
  result field = new Textfield (w);
  result field.setEditable (false);
  divideButton = new Button ("Devide"),
   devide Button.addActionlistener (this);
```

Add component to the frame

```
add (num1label);
add (num1field);
add (num2label);
add (num2field);
add (result label);
add (resultfield);
add (divideButton);
set visible (true);
}

public void actionperformed (ActionEvent e) {
    if (e.getsource() == devideButton) {
    try {
        int num1 = integer.parseInt (numfield.getText());
        int num2 = integer.parseInt (num1field.getText());
    if (num2 == 0) {
        showMessageDialog ("cannot divide by zero");
        return;
    }
    int result = num1/num2;
    result field.setText (string.valueof (result)); }
    catch (NumberformatException ex) {
        showMessage Dailog ( "please enter valid];
        integer for num1 and num2.");
    }
}
}

private void showMessageDialog (string Message){
    Dailog dialog = new Dailog (This, "Event", true);
    dailog setsize (400, 400);
    dailog setlayout (new flowlayout());
    label label = new label (Message);
    dalog add (label);
Button okButton = new Button ("ok");
OkButton.addActionListener(new Actionlistener()){
```

```
        public void actionperformed (ActionEvent e){
                dailog.dispose();
            }
        });
    dailog.add (okButton);
    dailog.setVisible (true);
  }
Public static void main (String args[]){
        new IntegerDivisionAWT();
    }
}
```

Output

NUM1: [20] NUM2: [5] ~~AWM~~ result: [4] [Divide]

9/2