

Jasmeet Singh (1BM22CS116)

Q1. Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
import java.lang.Math;
class quadratic
{public static void main(String XX[])
{
    int a,b,c;
    System.out.println("enter the values of a,b,c respectively\n");
    Scanner s1= new Scanner(System.in);
    a = s1.nextInt();
    b = s1.nextInt();
    c = s1.nextInt();
    double d= b*b - 4*a*c ;
    System.out.println("a = " + a + " b = " + b + " c = " + c);
    if(a==0) {System.out.println("not a quadratic equation");}
    else if( d>0)
    {
        System.out.println("the equation has two real and different solutions");
        double r1=(-b + Math.sqrt(d))/(2*a);
        double r2=(-b - Math.sqrt(d))/(2*a);
        System.out.println("r1 = " + r1);
        System.out.println("r2 = " + r2);
    }

    else if(d==0)
    {
        System.out.println("the equation has real and equal solutions");
        double r1= -b/(2*a);
        double r2= -b/(2*a);
        System.out.println("r1 = " + r1);
        System.out.println("r2 = " + r2);
    }

    else if(d<0)
    {
```

```
        System.out.println("the equation has unreal solutions");  
    }  
}  
}
```

Develop a java program that prints all real solutions to the quadratic eqⁿ $ax^2+bx+c=0$, Read in a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;  
import java.lang.Math;  
class quadratic;
```

```
public static void main(String XX[])
```

```
{  
    int a, b, c;
```

```
    System.out.println("Enter values of a, b, c respectively");
```

```
    Scanner s1 = new Scanner(System.in);
```

```
    a = s1.nextInt(); b = s1.nextInt();
```

```
    c = s1.nextInt();
```

```
    double d = b*b - 4*a*c;
```

```
    System.out.println("a = " + a + " b = " + b + " c = " + c);
```

```
    if (a == 0)
```

```
    { System.out.println("not a quadratic eqn"); }
```

```
    else if (d > 0)
```

```
    { System.out.println("the equation has two real  
    and different solutions");
```

```
    double x1 = (-b + Math.sqrt(d)) / (2*a);
```

```
    double x2 = (-b - Math.sqrt(d)) / (2*a);
```

```
    System.out.println("x1 = " + x1);
```

```
    System.out.println("x2 = " + x2);
```

```
    }  
    else if (d == 0)
```

```
    { System.out.println("the equation has real  
    and equal solutions");
```

Date / /
Page

```

double x1 = -b/(2*a);
double x2 = -b/(2*a);
system.out.println("x1 = " + x1);
system.out.println("x2 = " + x2);
}
else
{
    system.out.println("the equation has
        unreal solution");
}
}

```

OUTPUT

① enter the values of a, b, c respectively
 0 2 3
 a=0 b=2 c=3
 not a quadratic equation

② enter the values of a, b, c respectively
 1 5 2
 a=1 b=5 c=2
 the equation has two real and different solutions
~~x1 = -0.4384471~~
~~x2 = -4.5615528~~

③ enter the values of a, b, c respectively
 1 3 6
 a=1 b=3 c=6
 the equation has unreal solutions.

Q2. Develop a Java program to create a class Student with members usn, name, and array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;
class student{
    String USN , name;
    Scanner S1= new Scanner(System.in);
    int size = S1.nextInt();
    float credits[] = new float[size];
    float marks[] = new float[size];
    void accept(){
        USN= S1.next();
        name= S1.next();
        System.out.println("Marks in the following subjects are 1.Maths 2.physics
3.C prog 4.web 5.kannada 6.IDT 7.civil 8.english");
        for(int i=0;i<size;i++){
            System.out.print((i+1)+" = ");
            marks[i]= S1.nextInt();
        }
        System.out.println("respective credits of subjects are 1.Maths 2.physics
3.C prog 4.web 5.kannada 6.IDT 7.civil 8.english");
        for(int i=0;i<size;i++){
            System.out.print((i+1)+" = ");
            credits[i]= S1.nextInt();
        }
    }
    void display(){
        System.out.println("USN: "+USN+" name: "+name);
        System.out.println("Marks and credits in the following subjects are
1.Maths 2.physics 3.C prog 4.web 5.kannada 6.IDT 7.civil 8.english");
        for(int i=0;i<size;i++){
            System.out.print((i+1)+" .marks = "+marks[i]+" credits="+credits[i]+"
");
        }
    }
    int gpa(int i){
        if(marks[i]>=90) return 10;
        else if(marks[i]>=80 && marks[i]<=89) return 9;
        else if(marks[i]>=70 && marks[i]<=79) return 8;
        else if(marks[i]>=60 && marks[i]<=69) return 7;
```

```

        else if(marks[i]>=50 && marks[i]<=59) return 6;
        else if(marks[i]>=40 && marks[i]<=49) return 5;
        else if(marks[i]>=80 && marks[i]<=89) return 4;
        else return 0;
    }

    float sgpa(){
        float SGPA , sum=0;
        for(int i=0;i<size;i++){
            sum=sum+gpa(i)*credits[i];
        }
        //total credits=20
        SGPA = sum/20;
        return SGPA;
    }
}

class call{
    public static void main (String[] args) {
        student S1 = new student();
        S1.accept();
        S1.display();
        float Ans;
        Ans = S1.sgpa();
        System.out.println("SGPA of the student is : "+Ans);
    }
}

```

LAB-2

Date: 01/01/2024

Page

d) Develop a java program to create a class Student with members USN, name, an array credits and an array marks. Include methods to accept USN of a Student and a method to calculate

import java.util.Scanner

class Student {

String USN, name;

Scanner s1 = new Scanner(System.in);

int size = s1.nextInt();

float credits[] = new float[size];

float marks[] = new float[size];

void accept() {

USN = s1.next();

name = s1.next();

System.out.println("Marks in the following subjects
are 1. Maths 2. physics 3. c program 4. web
5. Kannada 6. IDT 7. civil 8. english");

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

System.out.print((i+1) + " = ");

marks[i] = s1.nextFloat();

System.out.println("respective credits of subjects
are 1. Maths 2. physics 3. c program 4. web
5. Kannada 6. IDT 7. civil 8. english");

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

System.out.print((i+1) + " = ");

credits[i] = s1.nextFloat();

}

void display() { for (int i = 0; i < size; i++)

{ System.out.print((i+1) + ". marks = " + marks[i]

+ " credits = " + credits[i] + " ");

}

int gpa(int i) {

```

if (marks[i] >= 90) return 10;
else if (marks[i] >= 80 && marks[i] <= 89)
    return 9;
else if (marks[i] >= 70 && marks[i] <= 79) return 8;
else if (marks[i] >= 60 && marks[i] <= 69) return 7;
else if (marks[i] >= 50 && marks[i] <= 59) return 6;
else if (marks[i] >= 40 && marks[i] <= 49) return 5;
else if (marks[i] >= 35 && marks[i] <= 40) return 4;
else return 0;
}

```

```

float sgpa() {
    float SGPA, sum = 0;
    for (int i = 0; i < size; i++) {
        sum = sum + gpa(i) * credit[i];
    }
    // total credits = 20
    SGPA = sum / 20;
    return SGPA;
}

```

```

class call {
    public static void main (String[] args)
    {
        Student s1 = new Student();
        s1.accept();
        s1.display();
        float Ans = s1.sgpa();
        System.out.println ("SGPA of the student
        is: " + Ans);
    }
}

```


Q3. Create a class Book which contains four members: name, author, price, 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.

```
import java.util.Scanner;
class Book{
    String name, Author;
    int price,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(){
        System.out.println("enter details of book :");
        Scanner S1= new Scanner(System.in);
        name = S1.next();
        Author = S1.next();
        price = S1.nextInt();
        num_pages= S1.nextInt();
    }
    void get(){
        System.out.println("name: "+name+" Author: "+Author+" price: "+price+"
num_pages: "+num_pages);
    }
    public String toString(){
        return("name: "+name+" Author: "+Author+" price: "+price+" num_pages:
"+num_pages);
    }
}
class bookdemo{
    public static void main (String[] args) {
        int n;
        Scanner S= new Scanner(System.in);
        System.out.print("print n: ");
        n=S.nextInt();
```

```
    Book B[] = new Book[n];
    B[0] = new Book("cant_hurt_me", "David_googins", 700, 360);
    B[0].get();
    for(int i=1; i<n; i++){
        B[i] = new Book();
        B[i].set();
    }
    System.out.println(B[1].toString());
    for(int i=2; i<n; i++){
        B[i].get();
    }
}
```

LAB-2

Date 01/01/24
Page _____

Q3 Create a class Book which contains four members: name, author, price, num-pages. Include a constructor to set the values of the members. Include methods to set and get the details of the object. Include a toString() method that could display the complete details of the book. Develop a java program to create n book objects.

```
import java.util.Scanner;

class Book {
    String name; Author;
    int price, 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() {
        System.out.println("Enter details of book: ");
        Scanner s1 = new Scanner(System.in);
        name = s1.next();
        Author = s1.next();
        price = s1.nextInt();
        num-pages = s1.nextInt();
    }
    void get() {
        System.out.println("Name: " + name + " Author: " + Author
            + " price: " + price + " num-pages: " + num-pages);
    }
}

class BookDemo {
    public String toString() {
        return "Name: " + name + " Author: " + Author
    }
}
```

Date: __/__/__
Page: __

```
+ "price: " + price + " Num. pages" + num_pages  
}  
}
```

```
class bookdemo {  
    public static void main (// String xx[])
```

```
    {  
        int n;
```

```
        Scanner S = new Scanner (System.in);
```

```
        System.out.println ("Enter n:");
```

```
        n = S.nextInt();
```

```
        Book B[] = new Book[n];
```

```
        B[0] = new Book ("cant read me", "David",  
            700, 360);
```

```
        System.out.println (B[0].toString());
```

```
        B[1] = new B
```

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

```
            B[i] = new Book();
```

```
            B[i].out();
```

```
            B[i].get();
```

```
        }  
    }
```

```
}
```


Q4. 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 only the method printArea() that prints the area of the given shape.

```
abstract class shape{
    int a,b;
    abstract void printarea();
}
class rectangle extends shape{
    rectangle(int x,int y){
        a=x;b=y;
    }
    void printarea(){
        System.out.println("area of rectangle is : "(a*b));
    }
}
class triangle extends shape{
    triangle(int x,int y){
        a=x;b=y;
    }
    void printarea(){
        System.out.println("area of triangle is : "(0.5*a*b));
    }
}
class circle extends shape{
    circle(int x){
        a=x;
    }
    void printarea(){
        System.out.println("area of circle is : "(3.14*a*a));
    }
}
class shapedemo{
    public static void main(String xx[]){
        rectangle r=new rectangle(5,4);
        triangle t=new triangle(5,4);
        circle c=new circle(7);
        r.printarea();
```

```
        t.printarea();  
        c.printarea();  
    }  
}
```

ques 5

Date ___/___/___
Page ___

Q)

a) 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 of the classes extends the class `shape`. Each of the classes contain only the method `printArea()` that prints the area of the given `shape`.

and

```
→ abstract class shape {  
    int a, b;  
    abstract void printArea();  
}
```

```
class Rectangle extends shape {  
    Rectangle(int x, int y) {  
        a = x; b = y;  
    }  
    void printArea() {  
        SOP("Area of rectangle is " + (a * b));  
    }  
}
```

```
class Triangle extends shape {  
    Triangle(int x, int y) {  
        a = x;  
        b = y;  
    }  
    void printArea() {  
        SOP("The Area of triangle is " + ((a * b) / 2));  
    }  
}
```

```
class circle extends shape {
    circle (int r) { a = r; }
    void printArea ( )
    {
        SOP ("area of circle is " + (3.14 * r * r));
    }
}
```

```
class ShapeDemo {
    public static void main (String XX[]) {
        Rectangle r = new Rectangle (5, 4);
        triangle t = new Triangle (5, 4);
        circle c = new circle (2);
        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

OUTPUT

→ area of Rectangle is 20
area of Triangle is 10
area of circle is

Q5. 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: a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;

class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;

    Account(String name, int accNo, String accType, double initialBalance) {
        customerName = name;
        accountNumber = accNo;
        accountType = accType;
        balance = initialBalance;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of $" + amount + " successful.");
    }

    void displayBalance() {
        System.out.println("Balance: $" + balance);
    }
}
```

```

    }
}

class CurAcct extends Account {
    double minBalance;
    double serviceCharge;

    CurAcct(String name, int accNo, String accType, double initialBalance, double
minBal, double charge) {
        super(name, accNo, accType, initialBalance);
        minBalance = minBal;
        serviceCharge = charge;
    }

    void withdraw(double amount) {
        if (balance - amount >= minBalance) {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount + " successful.");
        } else {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
    }

    void deductServiceCharge() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge of $" + serviceCharge + " applied
due to balance below minimum.");
        }
    }
}

class SavAcct extends Account {
    double interestRate;

    SavAcct(String name, int accNo, String accType, double initialBalance, double
interest) {
        super(name, accNo, accType, initialBalance);
        interestRate = interest;
    }

    void calculateInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest of $" + interest + " added.");
    }
}

```

```

    }

    void withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount + " successful.");
        } else {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
    }
}

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

        SavAcct savings = new SavAcct("John Doe", 123456, "Savings", 1000, 5);

        CurAcct current = new CurAcct("Jane Doe", 654321, "Current", 2000, 500, 10);

        System.out.println("Welcome to our bank!");

        while (true) {
            System.out.println("\n1. Deposit\n2. Withdraw\n3. Display Balance\n4.
Exit");

            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    System.out.print("Select account (1 for Savings, 2 for
Current): ");

                    int accountChoice = scanner.nextInt();
                    if (accountChoice == 1)
                        savings.deposit(depositAmount);
                    else if (accountChoice == 2)
                        current.deposit(depositAmount);
                    break;
                case 2:
                    System.out.print("Enter amount to withdraw: ");
                    double withdrawAmount = scanner.nextDouble();

```

```

        System.out.print("Select account (1 for Savings, 2 for
Current): ");

        accountChoice = scanner.nextInt();
        if (accountChoice == 1)
            savings.withdraw(withdrawAmount);
        else if (accountChoice == 2) {
            current.withdraw(withdrawAmount);
            current.deductServiceCharge();
        }
        break;
    case 3:
        System.out.print("Select account (1 for Savings, 2 for
Current): ");

        accountChoice = scanner.nextInt();
        if (accountChoice == 1)
            savings.displayBalance();
        else if (accountChoice == 2)
            current.displayBalance();
        break;
    case 4:
        System.out.println("Thank you for banking with us!");
        System.exit(0);
    default:
        System.out.println("Invalid choice. Please try again.");
}
}
}
}

```


MS thread
thread

Date / /
Page

a 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 withdraw 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 fall below, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the cur-account & sav-acc to make them more specific to their requirements. Include the necessary methods

- (a) Accept deposit from customer & update balance
- (b) display balance
- (c) compute and deposit interest
- (d) permit withdrawal & update the balance

```
import java.util.Scanner;  
class Account {  
    String customerName;  
    int accountNumber;  
    String accountType;  
    double balance;
```

```
    Account (String name, int accNo, String accType,  
             double initial balance)  
    {  
        customerName = name;  
        accountNumber = accNo;  
        accountType = accType;
```

```

balance = initialBalance;
}

void deposit (double amount)
{
    balance += amount;
    System.out.println("Deposit of ₹" + amount
        + " successful");
}

```

```

void displayBalance()
{
    SOP("Balance: ₹" + balance);
}

```

```

class CurAcct extends Account
{
    double minBalance;
    double serviceCharge;
    CurAcct (String name, int accNo, String
        acctype, double initialBalance, double min
        bal, double charge)
    {
        super (name, accNo, acctype, initialBalance);
        minBalance = minBal;
        serviceCharge = charge;
    }
}

```

```

void withdraw (double amount)
{
    if (balance - amount >= minBalance)
    {
        balance -= amount;
        SOP("Withdrawal of ₹" + amount +
            " successful");
    }
    else
    {
        SOP("Insufficient balance");
    }
}

```

```

void deductServiceCharge()
{
    if (balance < minBalance)
    {
        balance -= serviceCharge;
        SOP("Service charge of ₹" + serviceCharge
            + " applied due to balance below minimum");
    }
}

```

```

class SavAcct extends Account
{
    double interestRate;
    SavAcct (String name, int accNo, String acctype,
        double initialBalance, double interest)
    {
        super (name, accNo, acctype, initialBalance);
        interestRate = interest;
    }
}

```

```

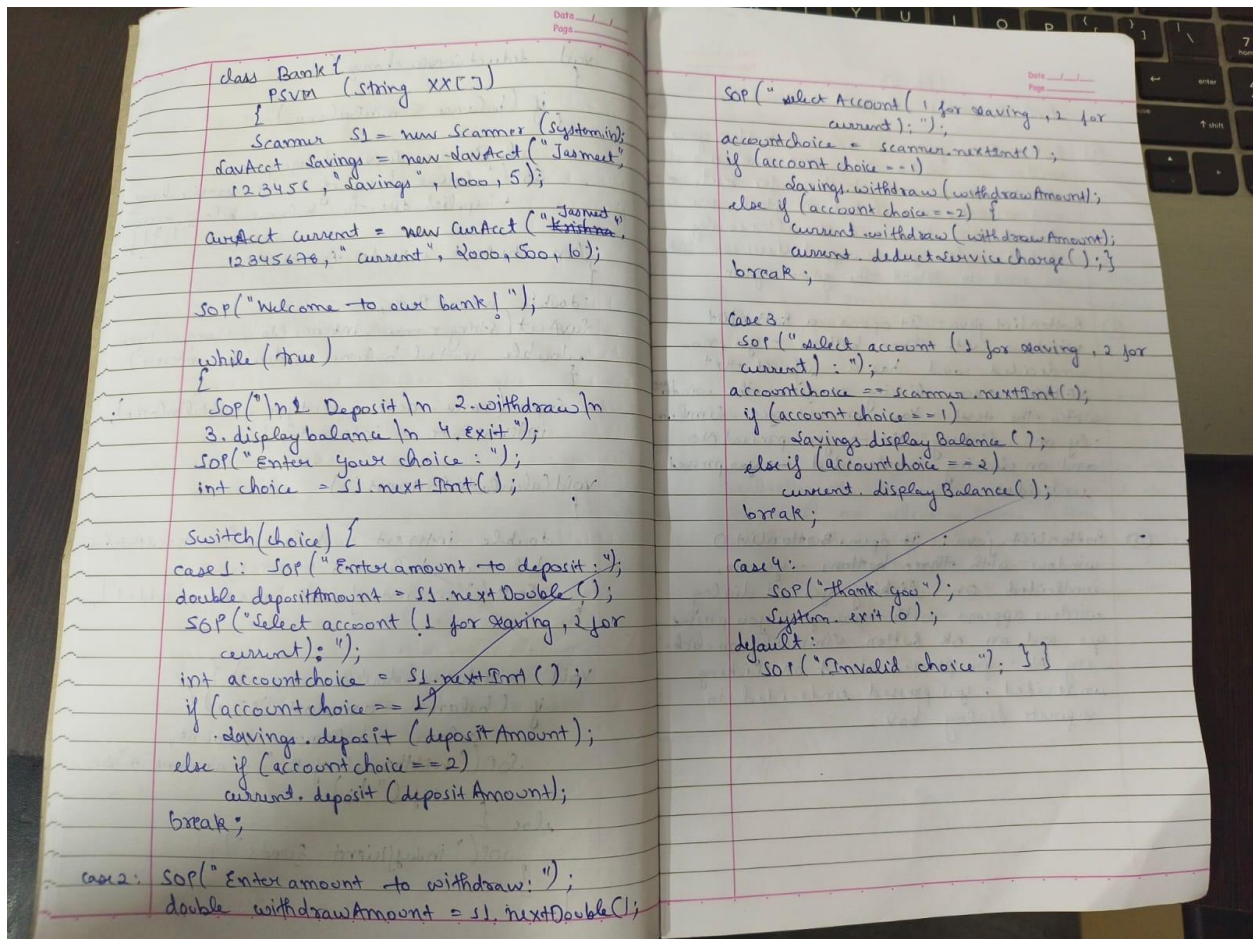
void calculateInterest()
{
    double interest = balance * interestRate / 100;
    balance = balance + interest;
    SOP("Interest of ₹" + interest + " added");
}

```

```

void withdraw (double amount)
{
    if (balance - amount >= 0)
    {
        balance = balance - amount;
        SOP("Withdrawal of ₹" + amount + " successful");
    }
    else
    {
        SOP("Insufficient funds");
    }
}

```

Q6. 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 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 in all five courses.

```

package CIE;
import java.util.Scanner;
public class student
{
    public String USN,name;
    public int semester;
    Scanner S1 = new Scanner(System.in);
    public void set()
    {
        System.out.println("USN of the student is : ");
        USN = S1.next();
        System.out.println("name of the student is : ");
        name = S1.next();
        System.out.println("semester of the student is : ");
        semester = S1.nextInt();
    }
    public void get()
    {
        System.out.println("USN of the student is : "+USN+"name of the student is : "+name+"semester of the student is : "+semester);
    }
}

```

```

package CIE;
import java.util.Scanner;
public class internal extends student
{
    Scanner S1 = new Scanner(System.in);
    public int internal_marks[]= new int[5];
    public void setcie()
    {
        System.out.println("1.Java 2.maths 3.DS 4.COA 5.DBMS ");
        for(int i=0;i<5;i++)
        {
            System.out.print("\n"+(i+1)+".");
            internal_marks[i]= S1.nextInt();
        }
    }
    public void getcie()
    {
        System.out.println("1.Java 2.maths 3.DS 4.COA 5.DBMS ");
        for(int i=0;i<5;i++)
        {
            System.out.println((i+1)+". "+internal_marks[i]);
        }
    }
}

```

```

    }
}
package SEE;
import CIE.student;
import java.util.Scanner;
public class external extends CIE.student
{
    Scanner s1 = new Scanner(System.in);
    public int see_marks[]=new int[5];
    public void setsee()
    {
        System.out.println("1.Java 2.maths 3.DS 4.COA 5.DBMS ");
        for(int i=0;i<5;i++)
        {
            System.out.print("\n"+(i+1)+".");
            see_marks[i]= s1.nextInt();
        }
    }
    public void getsee()
    {
        System.out.println(" 1.Java 2.maths 3.DS 4.COA 5.DBMS ");
        for(int i=0;i<5;i++)
        {
            System.out.println(i+"."+see_marks[i]);
        }
    }
}
}

```

```

import java.util.Scanner;
import CIE.student;
import CIE.internal;
import SEE.external;
public class fmarks
{
    public static void main(String XX[] )
    {
        int n;
        System.out.println("enter number of students : ");
        Scanner S1=new Scanner(System.in);
        n=S1.nextInt();
        CIE.student S[]=new student[n];
        CIE.internal I[]= new internal[n];
        SEE.external E[]= new external[n];
    }
}

```



```

for(int i=0;i<n;i++)
{
    S[i] = new student();
    I[i]= new internal();
    E[i]=new external();
    System.out.println("Enter details of student : ");
    S[i].set();
    System.out.println("Enter internal marks of student (out of 50) : ");
    I[i].setcie();
    System.out.println("Enter see marks of student (out of 100) : ");
    E[i].setsee();
}
for(int i=0;i<n;i++)
{
    System.out.println("details of student are : ");
    S[i].get();
    System.out.println("internal marks of student are : ");
    I[i].getcic();
    System.out.println("see marks of student are : ");
    E[i].getsee();
}
//final marks calculation
for (int i = 0; i < n; i++) {
    System.out.println("Marks of student " + S[i].name + ":");
    for (int j = 0; j < 5; j++) {
        int subjectTotalMarks = I[i].internal_marks[j] +
((E[i].see_marks[j])/2);
        System.out.println("Subject " + (j + 1) + " marks: " +
subjectTotalMarks);
    }
    System.out.println();
}

}
}

```

Date ___/___/___
Page ___

d Create a package CTE which has 2 classes student and Internal. The class student has members like USN, name, sem. The class Internal has an array that stores the internal marks scored in 5 courses of the current semester of the student. (create another package SEE which has the class External which is derived class of student. This class has an array that stores the SEE marks scored in 5 courses of current semester. Import the 2 packages in a file that declares the final marks of n students in all 5 courses.

→ Package CTE;

```
import java.util.Scanner;
public class student {
    public String name, USN;
    public int sem;
    public void accept() {
        Scanner S1 = new Scanner(System.in);
        name = S1.next();
        USN = S1.next();
        sem = S1.nextInt();
    }
    public void display() {
        SOP("Student details:\n" + "Name:"
            + name + "\nUSN: " + USN + "\nSemester: " + sem);
    }
}
```

```

package CIE
import java.util.Scanner;
public class Internal extends student {
    public int marks[] = new int[5];
    public void AcceptMarks() {
        Scanner s1 = new Scanner(System.in);
        for (int i=0; i<5; i++) {
            marks[i] = s1.nextInt();
        }
    }
    public void displayMarks() {
        for (int i=0; i<5; i++) {
            sop("Marks obtained in CIE  
subjects are " + (i+1) + " is "
            + marks[i]);
        }
    }
}

```

```

package SEE;
import CIE.student;
import java.util.Scanner;
public class External extends CIE.student {
    public int SEEmarks[] = new int[5];
    public void SEEAcceptmarks() {
        Scanner s1 = new Scanner(System.in);
        for (int i=0; i<5; i++) {
            SEEmarks[i] = s1.nextInt();
        }
    }
    public void SEEDisplayMarks() {
        for (int i=0; i<5; i++) {

```

```

        sop("Marks obtained in SEEs subjects "
        + (i+1) + " is " + SEEmarks[i]);
    }
}

```

```

import java.util.Scanner;
import CIE.student;
import CIE.internals;
import SEE.Externale;

```

```

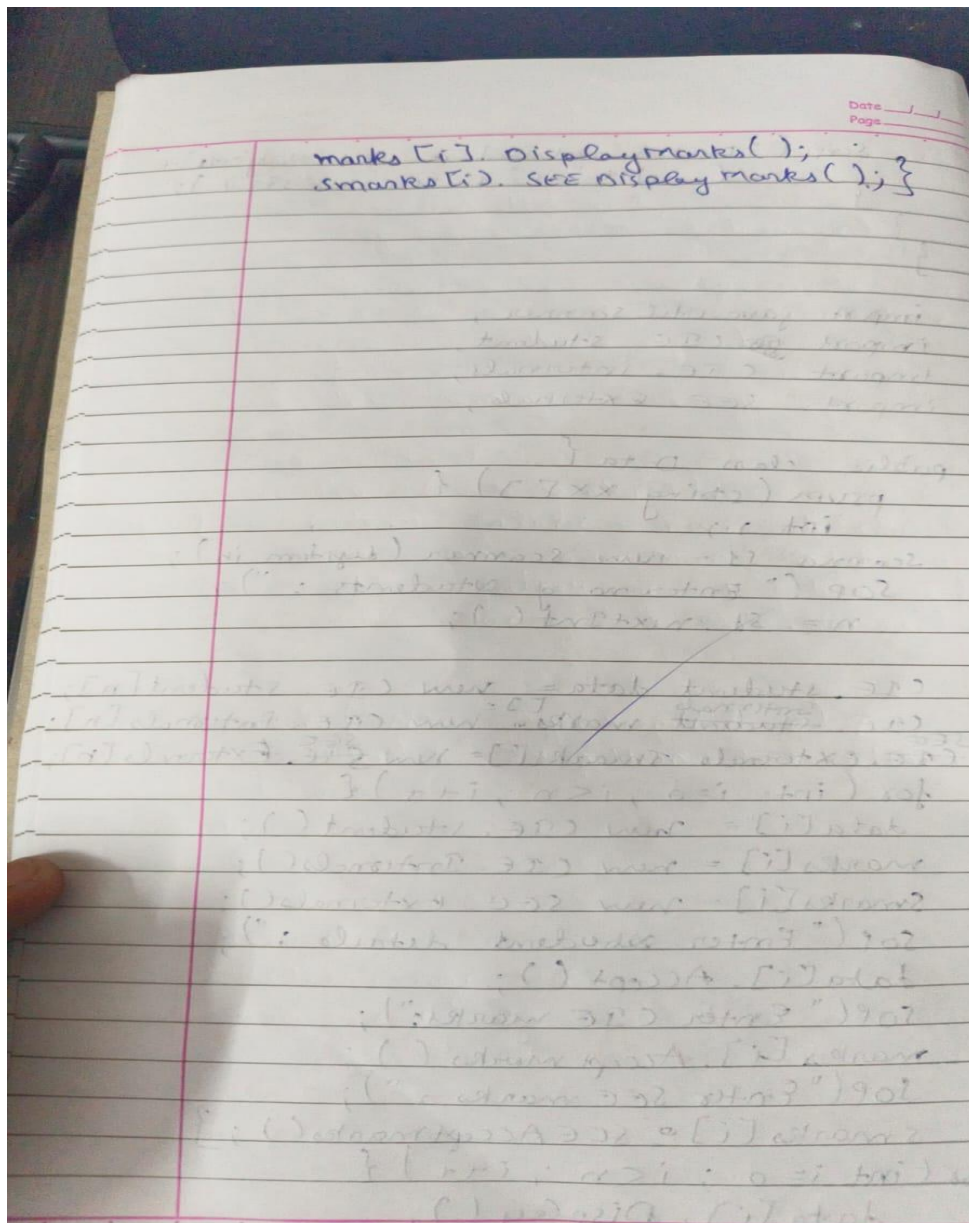
public class Data {
    psvm (String xx[]) {
        int n;
        Scanner s1 = new Scanner(System.in);
        sop("Enter no. of students: ");
        n = s1.nextInt();
    }

```

```

    CIE.student data = new CIE.student[n];
    CIE.internals marks = new CIE.internals[n];
    SEE.Externale smarks[] = new SEE.Externale[n];
    for (int i=0; i<n; i++) {
        data[i] = new CIE.student();
        marks[i] = new CIE.internals();
        smarks[i] = new SEE.Externale();
        sop("Enter student details:");
        data[i].Accept();
        sop("Enter CIE marks:");
        marks[i].Acceptmarks();
        sop("Enter SEE marks:");
        smarks[i].SEEAcceptmarks();
    }
    for (int i=0; i<n; i++) {
        data[i].Display();
    }
}

```

Q7. 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=father's age.

```

class wrongageexception extends Exception{
    wrongageexception(String message){
        super(message);
    }
    public String toString(){
        return "wrong age enetered";
    }
}
class father{
    int age;
    father(int age) throws wrongageexception{
        if(age<0){
            throw new wrongageexception("age cannot be negative");
        }
        this.age=age;
    }
}
class son extends father{
    int sonage;
    son(int fatherage, int sonage) throws wrongageexception{
        super(fatherage);
        if(sonage >= fatherage){
            throw new wrongageexception("son age cannot be greater than father
age");
        }
        this.sonage=sonage;
    }
}
class exceptiondemo{
    public static void main(String xx[])
    {
        try{
            father f=new father(40);
            son s=new son(f.age,25);
            System.out.println("father age :"+f.age);
            System.out.println("son age :"+s.sonage);
        }
        catch(wrongageexception e){
            System.out.println("exception:"+e.toString());
            System.out.println("exception:"+e.getMessage());
        }
    }
}

```


Q. Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called father class, implement a

```

→ import java.util.Scanner;

class WrongAgeException extends Exception
{
    WrongAgeException(String message) {
        super(message);
    }
}

class Father {
    int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Age cannot be negative");
        }
        this.age = age;
    }
}

class Son extends Father
{
    int sonage;

```

Date: / /
Page:

```

public son (int fatherage, int sonage)
throws WrongAgeException {
    super(fatherage);
    if (sonage >= fatherage)
    {
        throw new WrongAgeException("son
        Age cannot be greater than
        father's Age");
    }
    this.sonage = sonage;
}

```

```

class Main {
    public static void main (String xx[]) {
try {
    father father = new father(45);
    son son = new son(father.age, 20);
    sop("father age" + father.age);
    sop("sonage" + son.sonage);
}

```

```

catch (WrongAgeException e) {
    sop("Exception: + e.getMessage());
}
}

```

Output:

```

father's age: 45
son's age: 20

```

Q8. Write a program to create two threads. One thread displays "BMS college of Engineering" once every ten seconds and another displays "CSE" once every two seconds.

```
class BMSthread implements Runnable{
    public void run(){
        while(true){
            try{
                System.out.println("BMS College of engineering");
                Thread.sleep(10000);
            }
            catch(InterruptedException ie){
                System.out.println("Thread Interrupted");
            }
        }
    }
}

class CSEthread implements Runnable{
    public void run(){
        while(true){
            try{
                System.out.println("CSE");
                Thread.sleep(2000);
            }
            catch(InterruptedException ie){
                System.out.println("Thread Interrupted");
            }
        }
    }
}

class display{
    public static void main(String xx[]){
        thread bms=new thread(new BMSthread);
        thread cse=new thread(new CSEthread);
        bms.start();
        cse.start();
    }
}
```


Q. Write a program to create two threads one thread displays "BMS College of Engineering" once every ten seconds and another displays "CSE" once every two seconds.

→ class BMSThread implements Runnable {

public void Run() {

while (true)

{

try {

SOP("BMSCE");

Thread.sleep(10000);

}

catch (InterruptedException ie) {

SOP("BMS thread is interrupted");

}

}

}

class CSEThread implements Runnable {

public void Run() {

while (true) {

try {

SOP("CSE");

Thread.sleep(2000);

}

catch (InterruptedException ie) {

SOP("CSE thread is interrupted");

}

}

}

public class display {

public static void main (String [] args) {

```
- thread bms = new thread (new BMS thread  
- thread cse = new thread (new CSE thread  
  bms.start();  
  cse.start();
```

```
}
```

OUTPUT

BMS CSE

CSE

CSE

CSE

CSE

CSE

BMS CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

REPORT -

LAB-07

AWT programs - Report

- ① Button drag.java: It opens a button game window with 3 X 3 tiles with numbers and provides user with buttons for reset, swap and restart. One clicked on start, user can click on two tiles and simultaneously clicked tiles get swapped. User can click on reset to start the game again.
- ② ButtonList.java: It opens a ButtonList window with three buttons - yes, no, undecided and has a default text HELLO. Once clicking on yes, the window writes the text - You pressed yes. Similarly on clicking No → you pressed No and on clicking undecided → you pressed undecided.
- ③ ButtonList.java: It opens ButtonListD window with three buttons - yes, no, undecided. On clicking yes, a dialog window appears with message you pressed yes and an ok button. Similarly on clicking No: you pressed No & on clicking undecided, you pressed undecided in separate dialog box.
- ④ DivisonMain.java - It opens Divison of Integer window with fields to enter 2 numbers and a button RESULT. On clicking which the two numbers and quotient will appear on the window after Result. It gives and intakes numbers in float datatype.
- ⑤ DivisonMain.java - It opens a Divison of Integer window similar to last program. But the result provided will be integer numbers & the quotient in float.
- ⑥ TextFieldDemo.java - It opens a TF_labelDemo window with fields for entering name and password on clicking enter after entering the name, the text entered appears after Name. Similarly on clicking enter after the entering the password. It appears next to password. On selecting some letters in name field and clicking enter, the selected text appears next to select text in name. Password receives the input and displays it after encrypting it.

20/12/24