

```
import java.util.Scanner;
public class lab1 {
    public static void main(String[] args) {

        try (Scanner scanner = new Scanner(System.in)) {

            System.out.print("Enter coefficient a: ");
            double a = scanner.nextDouble();
            System.out.print("Enter coefficient b: ");
            double b = scanner.nextDouble();
            System.out.print("Enter coefficient c: ");
            double c = scanner.nextDouble();

            if (a == 0) {
                System.out.println("Coefficient a cannot be zero for a quadratic equation.");
                return;
            }
            double discriminant = b * b - 4 * a * c;

            if (discriminant > 0) {

                double sqrtDiscriminant = Math.sqrt(discriminant);
                double root1 = (-b + sqrtDiscriminant) / (2 * a);
                double root2 = (-b - sqrtDiscriminant) / (2 * a);
                System.out.println("The roots are real and different.");
                System.out.println("Root 1: " + root1);
                System.out.println("Root 2: " + root2);
            } else if (discriminant == 0) {

                double root = -b / (2 * a);
                System.out.println("The root is real and repeated.");
                System.out.println("Root: " + root);
            } else {

                System.out.println("There are no real solutions.");
            }
        }
    }
}
```

Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sanch>cd C:\Users\sanch\Desktop\java

C:\Users\sanch\Desktop\java>javac lab1.java

C:\Users\sanch\Desktop\java>java lab1

Enter coefficient a: 12

Enter coefficient b: 13

Enter coefficient c: 14

There are no real solutions.

C:\Users\sanch\Desktop\java>java lab1

Enter coefficient a: 2

Enter coefficient b: 1

Enter coefficient c: 0

The roots are real and different.

Root 1: 0.0

Root 2: -0.5

C:\Users\sanch\Desktop\java>java lab1

Enter coefficient a: 1

Enter coefficient b: 2

Enter coefficient c: 1

The root is real and repeated.

Root: -1.0

Done lot a Java program that finds all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use quadratic formula. If discriminant is negative, display there are no real solutions.

26/09/2020

VISION

Quadratic Equation

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

```
public class QuadraticSolver {
```

```
    public static void main(String[] args) {
```

```
        System.out.print("Enter coefficient a:");
```

```
        double a = scanner.nextDouble();
```

```
        System.out.print("Enter coefficient b:");
```

```
        double b = scanner.nextDouble();
```

```
        System.out.print("Enter coefficient c:");
```

```
        double c = scanner.nextDouble();
```

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

```
        if (discriminant > 0) {
```

```
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
```

```
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
```

```
            System.out.print("Roots are real  
and different");
```

```
}
```

```
else if (discriminant == 0) {
```

```
    double root = -b / (2 * a);
```

```
    System.out.print("Roots are real and  
same, root");
```

```
} else {
```

```
    double realPart = -b / (2 * a);
```

```
    double imaginaryPart = Math.sqrt(-discriminant) / (2 * a);
```

```
    System.out.print("Roots are complex and  
different in Root 1 = " + .2f + "j" + .2f + "i" + "
```

```
                    "Root 2 = " + .2f - .2f + "j" + .2f + "i" + ");
```

: Scanner · close()

Output:→ Enter Coefficient a: 12
Enter coefficient b: 13

Enter coefficient b: 13

Enter coefficient c : 14

Roots are complex and different

$$\text{Root 1} = -0.54 + 0.93i$$

$$\text{Root 2} = -0.54 \mp 0.93i$$

Enter coefficient a: 2

Enter coefficient b : 1

Ex: Roots are real and distinct
Root 1 = 0.00

$$\text{Root 2} = -0.50$$

Enter coefficient a: 1

~~Enter coefficient b: 2~~

Enter coefficient c : 1

~~Roots are real and same~~

$$\text{Root} = -1.0$$

~~26/10/24~~

```
import java.util.Scanner;
class Student {
    String usn;
    String name;
    int[] credits;
    int[] marks;

    public Student(int numSubjects) {
        credits = new int[numSubjects];
        marks = new int[numSubjects];
    }
    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter USN: ");
        usn = scanner.nextLine();

        System.out.print("Enter Name: ");
        name = scanner.nextLine();

        for (int i = 0; i < credits.length; i++) {
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            credits[i] = scanner.nextInt();
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
    }
    public void displayDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + " - Credits: " + credits[i] + ", Marks: " + marks[i]);
        }
    }
    public double calculateSGPA() {
        double totalCredits = 0;
        double totalGradePoints = 0;
        for (int i = 0; i < credits.length; i++) {
            double gradePoint = getGradePoint(marks[i]);
            totalGradePoints += gradePoint * credits[i];
            totalCredits += credits[i];
        }
        return totalCredits == 0 ? 0 : totalGradePoints / totalCredits;
    }
}
```

```
public double getGradePoint(int mark) {
    if (mark >= 90) return 10.0;
    else if (mark >= 80) return 9.0;
    else if (mark >= 70) return 8.0;
    else if (mark >= 60) return 7.0;
    else if (mark >= 50) return 6.0;
    else if (mark >= 40) return 5.0;
    else return 0.0; // Fail
}

public class lab2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter number of subjects: ");
        int numSubjects = scanner.nextInt();
        Student student = new Student(numSubjects);

        student.acceptDetails();
        student.displayDetails();

        double sgpa = student.calculateSGPA();
        System.out.printf("SGPA: %.2f\n", sgpa);
    }
}
```

Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sanch>cd C:\Users\sanch\Desktop\java

C:\Users\sanch\Desktop\java>javac lab2.java

C:\Users\sanch\Desktop\java>java lab2

Enter number of subjects: 5

Enter USN: 1BM23CS299

Enter Name: Sanchit

Enter credits for subject 1: 4

Enter marks for subject 1: 99

Enter credits for subject 2: 3

Enter marks for subject 2: 89

Enter credits for subject 3: 3

Enter marks for subject 3: 85

Enter credits for subject 4: 2

Enter marks for subject 4: 90

Enter credits for subject 5: 1

Enter marks for subject 5: 87

USN: 1BM23CS299

Name: Sanchit

Subject 1 - Credits: 4, Marks: 99

Subject 2 - Credits: 3, Marks: 89

Subject 3 - Credits: 3, Marks: 85

Subject 4 - Credits: 2, Marks: 90

Subject 5 - Credits: 1, Marks: 87

SGPA: 9.46

C:\Users\sanch\Desktop\java>java lab2

Enter number of subjects: 2

Enter USN: 1bm23cs123

Enter Name: xyz

Enter credits for subject 1: 1

Enter marks for subject 1: 78

Enter credits for subject 2: 5

Enter marks for subject 2: 98

USN: 1bm23cs123

Name: xyz

Subject 1 - Credits: 1, Marks: 78

Subject 2 - Credits: 5, Marks: 98

SGPA: 9.67

C:\Users\sanch\Desktop\java>java lab2

Enter number of subjects: 4

Enter USN: 1bm23cs189

Enter Name: piyush

Enter credits for subject 1: 4

Enter marks for subject 1: 80

Enter credits for subject 2: 3

Enter marks for subject 2: 87

Enter credits for subject 3: 2

Enter marks for subject 3: 92

Enter credits for subject 4: 1

Enter marks for subject 4: 67

USN: 1bm23cs189

Name: piyush

Subject 1 - Credits: 4, Marks: 80

Subject 2 - Credits: 3, Marks: 87

Subject 3 - Credits: 2, Marks: 92

Subject 4 - Credits: 1, Marks: 67

SGPA: 9.00

30/10/2024

P Develop a java program to create a class student with members, usn, name, an array credits and an array marks. Include methods to accept and display details.

import java.util.Scanner;

Class Student {

String USN;

String name;

int[] credits;

int marks;

void accepting details of student) {

Scanner ^{out} sc = new Scanner (System.in);

System.out.println ("Enter USN:");

USN = sc.nextLine();

System.out.print ("Enter name:");

name = sc.nextLine();

System.out.print ("Enter no of subjects:");

int n = sc.nextInt();

Credits = new int [n];

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

System.out.print ("Enter credits for sub" +
(i+1) + ":");

marks[i] = sc.nextInt();

}

void displayDetails () {

System.out.println ("USN: " + USN);

USN = sc.nextLine();

System.out.println ("Name: " + name);

System.out.println ("Credit and Marks: ");

```

for(int i=0; i < credits.length; i++) {
    System.out.println("Subject " + (i+1) + ":");
    credits = " " + credits[i] + " Marks = " +
    marks[i]);
}
}

```

```

double calculate SGPA() {
    int totalCredits = 0;
    double weightedSum = 0.0;
    for(int i=0; i < credits.length; i++) {
        weightedSum += marks[i] * credits[i];
        totalCredits += credits[i];
    }
    return weightedSum / totalCredits;
}

```

```

public static void main(String[] args) {
    Student student = new Student();
    student.acceptDetails();
    student.displayDetails();
    double sgpa = student.calculateSGPA();
    System.out.println("SGPA is " + sgpa);
}
}

```

Output: → Enter the number of students: 1

Enter USN: 13M23CS299

Enter Name: Sanchit Mehta

Enter number of subjects: 3

Enter the credits for subject 1: 4

Enter the marks for subject 1: 89

Enter the credits for subject 2: 3

Enter the marks for subject 2: 94

Enter credits for subject 3: 1

Enter the marks for subject 3: 97

USN : IBM23CS299

Name: Sanchit Mehta

Credits and marks:

Subject 1 : credits = 4 , marks = 89

Subject 2 : credits = 3 , marks = 94

Subject 1 : credits = 1 , marks = 97

SGPA : 9.8

N
3.0/100

```
import java.util.Scanner;

class Book {
    private String name;
    private String author;
    private double price;
    private int numPages;

    public Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

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

    public void setAuthor(String author) {
        this.author = author;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public void setNumPages(int numPages) {
        this.numPages = numPages;
    }

    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public int getNumPages() {
        return numPages;
    }

    public String toString() {
        return "Book Name: " + name + "\n" +
               "Author: " + author + "\n" +
               "Price: $" + price + "\n" +
               "Number of Pages: " + numPages;
    }
}
```

```
public class LAB3{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for book " + (i + 1) + ":");

            System.out.print("Name: ");
            String name = scanner.nextLine();

            System.out.print("Author: ");
            String author = scanner.nextLine();

            System.out.print("Price: ");
            double price = scanner.nextDouble();

            System.out.print("Number of Pages: ");
            int numPages = scanner.nextInt();
            scanner.nextLine();

            books[i] = new Book(name, author, price, numPages);
        }

        System.out.println("\nBook Details:");
        for (Book book : books) {
            System.out.println(book);
            System.out.println();
        }
        scanner.close();
    }
}
```

```
C:\Users\sanch\Desktop>java LAB3
Enter the number of books: 3
```

```
Enter details for book 1:
```

```
Name: The Alchemist
Author: Sanchit
Price: 7999.99
Number of Pages: 289
```

```
Enter details for book 2:
```

```
Name: Three men in a Boat
Author: Piyusg
Price: 101.99
Number of Pages: 34
```

```
Enter details for book 3:
```

```
Name: xyz
Author: abc
Price: 171.99
Number of Pages: 91
```

```
Book Details:
```

```
Book Name: The Alchemist
Author: Sanchit
Price: $7999.99
Number of Pages: 289
```

```
Book Name: Three men in a Boat
Author: Piyusg
Price: $101.99
Number of Pages: 34
```

```
Book Name: xyz
Author: abc
Price: $171.99
Number of Pages: 91
```

Q Create a class Book which contains 4 members : name, author, price, numPages. Include a constructor to set values for the members. Include the members to set and get the details of the objects. Include a toString() method that would display the complete details of the book. Develop a Java program to create 4 book objects.

Ans

```
import java.util.Scanner;
Class Book {
```

```
    private String name;
    private String author;
    private double price;
    private int numPages;
```

```
    public book (String name, String author,
double price; int int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
```

```
    public void setName (String name) {
        this.name = name;
    }
```

```
    public void setAuthor (String author) {
        this.author = author;
    }
```

```
    public void setPrice (double price) {
        this.price = price;
    }
```

```
public void setNumPages (int numPages) {  
    this.numPages = numPages;
```

```
}  
public String getName() {  
    return name;
```

```
}  
public String getAuthor() {  
    return author;
```

```
}  
public String getPrice() {  
    return price;
```

```
}  
public int getNumPages() {  
    return numPages;
```

```
}  
public String toString() {
```

```
    return "Book Name:" + name + " in " + "Author:"  
    + author + " in " + "Price:" + price + " in " + "Number of  
    Pages:" + numPages;
```

```
}  
public class Main2 {
```

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

```
Book[] books = new Book[m];  
  
for (int i=0; i<m; i++) {  
    System.out.println("Enter details for  
book " + (i+1) + ":");  
    System.out.print("Name: ");  
    String name = scanner.nextLine();  
  
    System.out.print("Author: ");  
    String author = scanner.nextLine();  
  
    System.out.print("Price: ");  
    double price = scanner.nextDouble();  
  
    System.out.print("No. of Pages: ");  
    int numPages = scanner.nextInt();  
  
    scanner.nextLine();  
  
    books[i] = new Book(name, author, price,  
        numPages);  
}
```

```
System.out.println("Book Details:");  
for (Book book : books) {  
    System.out.println(book);  
    System.out.println();  
}  
}
```

Output → Enter number of books: 3

Enter details for book 1:

Name: The Alchemist

Author: Sandeep

Price: 7999.99

No. of Pages: 289

Enter details for book 2:

Name: Three men in a Boat

Author: Piyush Singh

Price: 101.99

No. of Pages: 34

Enter details for book 3:

Name: xyz

Author: abc

Price: 171.99

No. of Pages: 91

Q 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 contains only method `printArea()` that prints the area of the given `shape`.

Ans

abstract class `Shape`

protected int `dimension1`;

protected int `dimension2`;

~~this.dimension1~~

public `Shape` (int `dimension1`, int `dimension2`)

{

 this.`dimension1` = `dimension1`;

 this.`dimension2` = `dimension2`;

}

} public abstract void `printArea()`;

Class `Rectangle` extends `Shape`

public `Rectangle` (int `width`, int `height`)

{

 super(`width`, `height`);

}

```
public void printArea() {  
    int area = dimension1 * dimension2;  
    System.out.println("Area of Rectangle: " + area);  
}
```

Class Triangle extends Shape {

```
public Triangle extends Shape {
```

```
    public Triangle (int base, int height) {
```

```
        Super(base, height);
```

```
    public void printArea() {
```

```
        double area = 0.5 * dimension1 * dimension2;
```

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

 }

Class Circle extends Shape {

```
    public Circle (int radius) {
```

```
        Super(radius, 0);
```

 }

```
    public void printArea () {
```

```
        double area = Math.PI * dimension1 *
```

```
                        dimension1;
```

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

 }

Ques. public class lab4 {

```
public static void main(String[] args)
{
    Rectangle
    Shape↑ = new Rectangle(5, 10);
    rectangle.printArea();
    rectangle.printArea();
    Shape↑ triangle = new Triangle(5, 10);
    triangle.printArea();
}
```

```
Shape circle = new Circle(7);
circle.printArea();
}
```

Output:- Area of Rectangle : 50

Area of Triangle : 25.0

Area of Circle : 153.93804002589

✓ 24/10/21

```
abstract class Shape {
    int dimension1;
    int dimension2;
    Shape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }
    public abstract void printArea();
}
class Rectangle extends Shape {
    public Rectangle(int width, int height) {
        super(width, height);
    }
    public void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}
class Triangle extends Shape {
    public Triangle(int base, int height) {
        super(base, height);
    }
    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}
class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0);
    }
    public void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}
public class lab4 {
    public static void main(String[] args) {
        Shape rectangle = new Rectangle(5, 10);
        rectangle.printArea();

        Shape triangle = new Triangle(5, 10);
        triangle.printArea();

        Shape circle = new Circle(7);
        circle.printArea();
    }
}
```

```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\sanch>cd C:\Users\sanch\Desktop\java
```

```
C:\Users\sanch\Desktop\java>javac lab4.java
```

```
C:\Users\sanch\Desktop\java>java lab4
Area of Rectangle: 50
Area of Triangle: 25.0
Area of Circle: 153.93804002589985
```

7/1/24

VISION

Date

VISION

Page No.

P Develop a Java program to create a class Bank that maintains two kinds of accounts 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 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 Curr-acct and Sav-acct to make them more specific to their requirements.

Include the necessary methods / member to achieve the following task:

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

class

import java.util.Scanner

Class Account {

```
String CustomerName,  
String typeAccount;  
Double balance;  
long double accnum;
```

```
Account(String CustomerName, String typeAccount,  
double balance, long accnum)
```

}

```
this.CustomerName = CustomerName;  
this.typeAccount = typeAccount;  
this.balance = balance;  
this.accnum = accnum;
```

}

public void deposit(double amount)

{

```
if (amount > 0)
```

{

```
balance += amount;
```

}

else

{

```
System.out.println("Invalid Amount");
```

}

public void display()

{
 System.out.println("The balance is " + balance);
}

Class Savacc extends Account {

 double interestRate;

Savacc (String CustomerName, double balance, long
 accnum, double interestRate)

{
 super (CustomerName, balance, accnum,
 interestRate);

 this.interestRate = interestRate;

}

public CompoundInterest()

{

 double interest = balance * (interestRate / 100);
 deposit (interest);

}

 System.out.println ("Interest Compound = " +
 interest);

public withdraw (double amount)

{

 if (amount > 0 & & amount <= balance)

{

 balance -= amount;

}

```
else
{
    System.out.println("Invalid amount");
}
```

class Current extends Account

```
{  
    final double MinBal = 4000;  
    final double Service_charge = 100;
```

public void withdraw (double amount)

```
{  
    if (amount > 0 && amount <= balance)  
    {  
        balance -= amount;  
        System.out.println("withdraw" +  
                           amount);  
    }  
}
```

checkMinBalance();

public void checkMinBalance()

```
{  
    else  
    {  
        System.out.println("Invalid  
                           amount");  
    }  
}
```

```
public void checkMinBalance() {
```

```
    if (balance < Min_Bal)
```

```
}
```

```
    balance -= Service_charge;
```

```
    System.out.println("Minimum Balance is not  
    maintained");
```

```
}
```

```
} }
```

```
public class Bank {
```

```
    public static void main(String[] args) {
```

```
        SavAcc savings = new SavAcc("Anur", 1234567911L,  
        5500, 9.4);
```

```
        savings.display();
```

```
        savings.deposit(1000);
```

```
        savings.display();
```

```
        savings.withdrawal(500);
```

```
        savings.display();
```

```
        savings.compoundInterest();
```

```
        savings.display();
```

```
        System.out.println("Current account operations")
```

CurAcc current = new CurAcc ("Sanchit",
1234567899L, 2000),

current.display();

current.deposit(500);

current.display();

current.withdraw(1200);

current.display();

}

}

Output: → The balance is: 5500

Deposited : 1000

The balance is: 6500

Withdraw : 500

The balance is: 6000

Deposited : 564

~~The~~ Interest compounded : 564

The balance is: 6564

✓ 28/11/29
VS

Current account operations

The balance is: 2000

Deposited : 500

The balance is: 2500

Withdraw : 1200

The balance is: 1300

```
import java.util.Scanner;

class Account {
    String customerName;
    String accountNum;
    double balance;

    public Account(String customerName, String accountNum, double balance) {
        this.customerName = customerName;
        this.accountNum = accountNum;
        this.balance = balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Invalid amount");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: " + amount);
        } else {
            System.out.println("Invalid request or insufficient funds");
        }
    }

    public void display() {
        System.out.println("The balance is: " + balance);
    }
}

class SavAcc extends Account {
    double interestRate;

    public SavAcc(String customerName, String accountNum, double balance, double interestRate) {
        super(customerName, accountNum, balance);
        this.interestRate = interestRate;
    }
}
```

```
public void compoundInterest() {
    double interest = balance * (interestRate / 100);
    deposit(interest);
    System.out.println("Interest compounded: " + interest);
}
}

class CurAcc extends Account {
    static final double MIN_BAL = 1000;
    static final double SERVICE_CHARGE = 100;

    public CurAcc(String customerName, String accountNum, double balance) {
        super(customerName, accountNum, balance);
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: " + amount);
            checkMinBalance();
        } else {
            System.out.println("Invalid request or insufficient balance");
        }
    }

    public void checkMinBalance() {
        if (balance < MIN_BAL) {
            balance -= SERVICE_CHARGE;
            System.out.println("Minimum balance not maintained. Service charge imposed: " + SERVICE_CHARGE);
        }
    }
}

public class lab5 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter customer name for savings account: ");
        String savName = scanner.nextLine();
        System.out.print("Enter account number for savings account: ");
        String savAccNum = scanner.nextLine();
        System.out.print("Enter initial balance for savings account: ");
        double savBalance = scanner.nextDouble();
        System.out.print("Enter interest rate for savings account: ");
        double savInterestRate = scanner.nextDouble();

        scanner.nextLine();
    }
}
```

```
SavAcc savings = new SavAcc(savName, savAccNum, savBalance, savInterestRate);
savings.display();

System.out.print("Enter the amount to deposit in savings account: ");
savings.deposit(scanner.nextDouble());
savings.display();

System.out.print("Enter the amount to withdraw from savings account: ");
savings.withdraw(scanner.nextDouble());
savings.display();

savings.compoundInterest();
savings.display();

System.out.print("\nEnter customer name for current account: ");
String currName = scanner.nextLine();
System.out.print("Enter account number for current account: ");
String currAccNum = scanner.nextLine();
System.out.print("Enter initial balance for current account: ");
double currBalance = scanner.nextDouble();

scanner.nextLine();

CurAcc current = new CurAcc(currName, currAccNum, currBalance);
current.display();

System.out.print("\nEnter the amount to deposit in current account: ");
current.deposit(scanner.nextDouble());
current.display();

System.out.print("Enter the amount to withdraw from current account: ");
current.withdraw(scanner.nextDouble());
current.display();

scanner.close();
}
```

```
C:\1bm23cs299>java lab5
Enter customer name for savings account: sanchit
Enter account number for savings account: 111
Enter initial balance for savings account:
23
Enter interest rate for savings account: 12
The balance is: 23.0
Enter the amount to deposit in savings account: 222
Deposited: 222.0
The balance is: 245.0
Enter the amount to withdraw from savings account: 21
Withdrew: 21.0
The balance is: 224.0
Deposited: 26.88
Interest compounded: 26.88
The balance is: 250.88

Enter customer name for current account: Enter account number for current account: xyz
Enter initial balance for current account: 1111
The balance is: 1111.0

Enter the amount to deposit in current account: 125678
Deposited: 125678.0
The balance is: 126789.0
Enter the amount to withdraw from current account: 2223
Withdrew: 2223.0
The balance is: 124566.0
```

Q Create a package CIE which has two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals 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}Personal. This class has an array that stores the SEE marks stored in 5 courses of current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

Ans package cie;

public class Internals {

int[] marks = new int[5];

public Internals (int[] marks)

{

this.marks = marks;

}

public void display () {

System.out.println ("CIE Marks:");

for (int num : marks) {

System.out.println (num);

}

public int getTotalCIEMarks () {

int total = 0;

for (int num : marks) {

total += num;

}

} } return total;

package see;

import cie. student;

public class External extends Student {

int[] seemarks = new int[5];

public External (String USN, String name,
int sem, int[] seemarks)

{

super (USN, name, sem);

this. seemarks = seemarks;

}

public void display () {

System.out.println ("SEE marks are : " + num);

for (int num : seemarks)

{

System.out.println (num);

}

~~public class~~ ~~see~~ ~~int~~ public int gethalf () {

int total = 0;

for (int num : seemarks)

{

total += num;

}

return total / 2;

{

package cie;

public class Student {

 public String USN;

 public String name;

 public int sem;

 public Student (String USN, String name,
 int sem) {

 this.USN = USN;

 this.name = name;

 this.sem = sem;

}

 public void display () {

 System.out.println ("Student Details:");

 System.out.println ("USN: " + USN);

 System.out.println ("Name: " + name);

 System.out.println ("Semester: " + sem);

}

```
import java.util.Scanner  
import file.Student  
import file.Internals  
import file.External
```

public class Lab69

public static void main(String[] args)

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

```
System.out.println("Enter number of  
Students : ");
```

```
int n = s.nextInt();
```

s.nextLine();

```
for( int i=1; i<=n; i++ )
```

```
- System.out.println ("Enter details  
of student " + i + ":"),
```

```
System.out.println("Enter USN:");
String USN = s.nextLine();
```

```
System.out.println("Enter name:");  
String name = s.nextInt();  
s.nextLine();
```

System.out.println ("Enter CIE marks for 5 subjects : ");

int [] smarks = new int [5];

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

{ smarks [j] = s.nextInt(); }

External externalmarks = new External (USN, name, sem, smarks);

externalmarks.display();

s.nextLine();

}

s.close();

}

~~Output~~ → Enter number of students:

2

Enter details of Student 1:

Enter USN:

1bm23cs222

~~Name~~ Enter Name:

xyz

Enter semester:

3

Student Details:
USN: 1bm23cs 222

Name: xyz

Semester: 3

Enter CIE marks for 5 subjects:

45

46

47

48

49

CIE Marks:

45

46

47

48

49

Enter the SEE marks for 5 subjects:

89

90

96

99

78

SEE marks are as follows:

89

90

96

99

78

Student details for Student 2:
Enter USN:

1bm23cs123

Enter Name:

abc

Enter Semester:

3

Student Details:

USN: 1bm23cs123

Name: abc

Semester: 3

Enter CIE marks for 5 subjects:

50

45

34

23

48

CIE Marks:

50

45

34

23

48

Enter SEE Marks for 5 subjects:

90

92

88

78

49

SEE marks are as follows:

90
92
88
78
~~96~~

✓ 28/1/24
VS

```
package cie;
public class Internals {
    int[] marks = new int[5];

    public Internals(int[] marks) {
        this.marks = marks;
    }

    public void display() {
        System.out.println("CIE Marks: \n");
        for (int num : marks) {
            System.out.println(num);
        }
    }

    public int getTotalCieMarks() {
        int total = 0;
        for (int num : marks) {
            total += num;
        }
        return total;
    }
}
```

```
package cie;

public class Student {
    public String USN;
    public String name;
    public int sem;

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

    public void display() {
        System.out.println("Student Details:");
        System.out.println("USN: " + USN);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
```

```
package see;
import cie.Student;

public class External extends Student {
    int[] seemarks = new int[5];

    public External(String USN, String name, int sem, int[] seemarks) {
        super(USN, name, sem);
        this.seemarks = seemarks;
    }

    public void display() {
        System.out.println("SEE Marks are: \n");
        for (int num : seemarks) {
            System.out.println(num);
        }
    }

    public int getHalfOfSeeMarks() {
        int total = 0;
        for (int num : seemarks) {
            total += num;
        }
        return total / 2;
    }
}
```

```
import java.util.Scanner;
import cie.Student;
import cie.Internals;
import see.External;

public class lab6 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);

        System.out.println("Enter the number of students:");
        int n = s.nextInt();
        s.nextLine();

        for (int i = 1; i <= n; i++) {
            System.out.println("\nEnter details for Student " + i + ":");

            System.out.println("Enter USN: ");
            String USN = s.nextLine();

            System.out.println("Enter Name: ");
            String name = s.nextLine();

            System.out.println("Enter Semester: ");
            int sem = s.nextInt();
            s.nextLine();

            Student stud = new Student(USN, name, sem);
            stud.display();

            int[] marks = readMarks(s, "CIE");
            Internals internalMarks = new Internals(marks);
            internalMarks.display();

            int[] smarks = readMarks(s, "SEE");
            External externalMarks = new External(USN, name, sem, smarks);
            externalMarks.display();

            int[] finalMarks = new int[5];
            System.out.println("Final Marks (CIE + 1/2 SEE) for each subject:");
            for (int j = 0; j < 5; j++) {
                finalMarks[j] = marks[j] + (smarks[j] / 2);
                System.out.println("Subject " + (j + 1) + ": " + finalMarks[j]);
            }
        }

        s.close();
    }

    public static int[] readMarks(Scanner s, String examType) {
        System.out.println("Enter the " + examType + " marks (5 subjects): ");
        int[] marks = new int[5];
        for (int i = 0; i < 5; i++) {
            marks[i] = s.nextInt();
        }
        return marks;
    }
}
```

```
Enter the number of students: 2
Enter details for student 1:
USN: 1bm23
Name: sam
Semester: 3
Enter internal marks for 5 courses:
30 39 29 28 30
Enter SEE marks for 5 courses:
86 87 89 90 99
Enter details for student 2:
USN: 1bm24
Name: xyz
Semester: 2
Enter internal marks for 5 courses:
39 34 35 36 34
Enter SEE marks for 5 courses:
99 98 87 86 80
```

```
Final Marks of Students:
USN: 1bm23
Name: sam
Semester: 3
Internal Marks: 30 39 29 28 30
SEE Marks: 86 87 89 90 99
Final Marks: 116 126 118 118 129

USN: 1bm24
Name: xyz
Semester: 2
Internal Marks: 39 34 35 36 34
SEE Marks: 99 98 87 86 80
Final Marks: 138 132 122 122 114
```

21/11/24

Date

Page No.

Q Write a Program that demonstrates handling of exceptions in inheritance tree. Create a base class called 'Father' and a 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 the Son class, implement a constructor that uses both father and son's age and throws an exception if son's age is \geq father's age.

Ans Class WrongAgeException Extends Exception {

 public String toString() {

 return "WrongAgeException";

Class Father {

 int age;

 Father(int age) throws WrongAgeException {
 if (age < 0)
 throw new WrongAgeException();

 }

 this.age = age;

}

```
int getAge()
```

```
    return age;
```

Class Son extends Father

```
{ int getSonAge();
    Son (int fatherAge, int sonAge) knows
        WrongAgeException {
            super (fatherAge);
            if (sonAge < 0 || sonAge > = fatherAge) {
                throw new WrongAgeException();
            }
            this . sonAge = sonAge;
        }
}
```

```
int getSonAge ()
```

```
    return sonAge;
```

public class ExceptionHandlingInheritance {

```
    public static void main (String [] args) {
        try {
            Father f1 = new
                Father (40);
```

```
System.out.println ("Father's age : " + f1.getAge());
```

```
Son s1 = System.out.println ("Enter father's age : ");
try {
    int fatherAge = scanner.nextInt();
```

```
Father f1 = new Father (fatherAge);
System.out.println ("Father's age : " +
    f1.getAge());
```

```
System.out.print ("Enter Son's age : ");
int sonAge = scanner.nextInt();
```

```
Son s1 = new Son (fatherAge, sonAge);
System.out.println ("Son's Age : " +
    s1.getSonAge());
```

```
Catch (WrongAgeException e) {
    System.out.println ("Exception
        Caught : " + e);
```

Output Enter father's age : 12

Father's age : 12

Enter son's age : 34

Exception Caught : WrongAgeException

Enter Father's age : 12

Father's age : 12

Enter son's age : -23

Exception Caught : WrongAgeException

Problems -

Chances

Page No.

VISION -

Date

Page No.

Enter father's age: -33

Exception Caught: WrongAge Exception

✓ 3/12/2011 29

```
import java.util.Scanner;
class WrongAgeException extends Exception {
    public String toString() {
        return "WrongAgeException";
    }
}
class Father {
    int age;
    Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException();
        }
        this.age = age;
    }
    int getAge() {
        return age;
    }
}
class Son extends Father {
    int sonAge;
    Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0 || sonAge >= fatherAge) {
            throw new WrongAgeException();
        }
        this.sonAge = sonAge;
    }
    int getSonAge() {
        return sonAge;
    }
}
public class lab7 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        try {
            System.out.print("Enter Father's age: ");
            int fatherAge = scanner.nextInt();

            Father f1 = new Father(fatherAge);
            System.out.println("Father's age: " + f1.getAge());
            System.out.print("Enter Son's age: ");
            int sonAge = scanner.nextInt();

            Son s1 = new Son(fatherAge, sonAge);
            System.out.println("Son's age: " + s1.getSonAge());
        } catch (WrongAgeException e) {
            System.out.println("Exception caught: " + e);
        } finally {
            scanner.close();
        }
    }
}
```

```
C:\1bm23cs299>java lab7
Enter Father's age: 12
Father's age: 12
Enter Son's age: 34
Exception caught: WrongAgeException

C:\1bm23cs299>java lab7
Enter Father's age: 12
Father's age: 12
Enter Son's age: -23
Exception caught: WrongAgeException

C:\1bm23cs299>java lab7
Enter Father's age: -33
Exception caught: WrongAgeException
```

- Q) Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every 10 seconds and other displaying "CSE" every two seconds.

Class threads extends Thread

String s;

int time;

threads (String s, int time)

{

this.s = s;

this.time = time;

}

public void run () {

try {

while (true)

{

System.out.println(s);

Thread.sleep (time * 1000);

}

Catch (Exception e) {

System.out.println(e);

}

}

public class Main {

 public static void main (String args)

 {
 threads t₁ = new threads (

 " BMS College of Engineering",
 1);

 threads t₂ = new threads ("CSE", 2);

 t₁. start();

 t₂. start();

}

Output : BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

N
28/11/24

```
class threads extends Thread{
    String s; int time;
    threads(String s, int time){
        this.s = s;
        this.time = time;
    }
    public void run(){
        try{
            while(true){
                System.out.println(s);
                Thread.sleep(time * 1000);
            }
        } catch(Exception e){
            System.out.println(e);
        }
    }
}
public class lab8{
    public static void main(String[] args) {
        threads t1 = new threads("BMS College of Engineering", 10);
        threads t2 = new threads("CSE", 2);

        t1.start();
        t2.start();
    }
}
```

```
C:\1bm23cs299>java lab8
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
```