

1.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;

class lab1{
    public static void main(String args[]){
        double a;
        double b;
        double c;
        Scanner sx = new Scanner(System.in);
        System.out.print("enter value a : ");
        a = sx.nextDouble();
        System.out.print("enter value b : ");
        b = sx.nextDouble();
        System.out.print("enter value c : ");
        c = sx.nextDouble();

        double discriminant = b*b-(4*a*c);
        if(a == 0){
            System.out.println("it is not a quadratic equation");
        }
        else{
            if(discriminant<0){
                System.out.println("no real roots...");
            }
            else if(discriminant == 0){
                double root = -b/(2*a);
                System.out.println("there is one real root "+root);
            }
            else{
                double sqrtdiscriminant = Math.sqrt(discriminant);
```

```
        double root1 = (-b+sqrtdiscriminant)/(2*a);
        double root2 = (-b-sqrtdiscriminant)/(2*a);
        System.out.println("real roots are "+root1+" and "+root2);
    }
}
}
}
}
```

O/P:

```
PS D:\3rd sem\JAVA\new practice> javac lab1.java
PS D:\3rd sem\JAVA\new practice> java lab1
enter value a : 1
enter value b : -5
enter value c : 6
real roots are 3.0 and 2.0
PS D:\3rd sem\JAVA\new practice> java lab1
enter value a : 1
enter value b : 3
enter value c : 4
no real roots...
```

1. 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;

public class Quadratic {

public static void main (String args[]) {

int a, b, c;

Scanner sx = new Scanner (System.in);

System.out.print ("enter the value of a : ");
a = sx.nextInt();

System.out.print ("enter the value of b : ");
b = sx.nextInt();

System.out.print ("enter the value of c : ");
c = sx.nextInt();

sx.close();

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

if (discriminant <= 0) {

System.out.println ("there are no real
solutions...!");

else {

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

double quad1 = (-b - Math.sqrt(discriminant))
/ (2 * a);

System.out.println ("1st Root : " + quad +
" 2nd Root : " + quad1);

y
y
y

O/P: 2.00
if (discriminant == 0)
{
 system.out.println("root 1 = " + (-b / (2.0
 * a))) ;
}

→ enter : a : 2

enter : b : 3

enter : c : 4.

no real solutions ... !

→ enter : a : 3

Enter b : -7

enter c : -3

root 1 : 2.70

root 2 : -0.36

2: 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 and a method to calculate SGPA of a student.

```
import java.util.Scanner;

class student{
    String usn;
    String name;
    int []credits;
    int []marks;
    double sgpa ;
    void putd(){
        Scanner sx = new Scanner(System.in);
        System.out.print("enter usn : ");
        usn = sx.next();
        System.out.print("enter name : ");
        name = sx.next();
        int n;
        System.out.print("enter number of subjects : ");
        n = sx.nextInt();
        credits = new int[n];
        marks = new int[n];
        for(int i = 0;i<n;i++){
            System.out.print("enter marks in subject "+(i+1)+" : ");
            marks[i] = sx.nextInt();
            System.out.print("enter credits of subject "+(i+1)+" : ");
            credits[i] = sx.nextInt();
            System.out.println();
        }
        int totalCredits = 0;
        double Weightedgradepoints = 0;
        for(int i = 0;i<n;i++){
```

```

        double gradepoint = marks[i] / 10.0 ;
        Wightedgradepoints += gradepoint * credits[i];
        totalCredits += credits[i];
    }
    sgpa = Wightedgradepoints/totalCredits;
}
void display()
{
    System.out.println("your SGPA : "+sgpa);
}
}

class lab2{
    public static void main(String args[]){
        student s1 = new student();
        s1.putd();
        s1.display();
    }
}

```

O/P :

```

enter usn : 1BM24CS427
enter name : uday_shankar_y
enter number of subjects : 4
enter marks in subject 1 : 89
enter credits of subject 1 : 3

enter marks in subject 2 : 78
enter credits of subject 2 : 44

enter marks in subject 3 : 80
enter credits of subject 3 : 2

enter marks in subject 4 : 76
enter credits of subject 4 : 4

your SGPA  : 7.854716981132075

```

2 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 and details and a method to calculate CGPA of a student.

→

```
import java.util.Scanner;
class Stu {
    int max_credits[] = {4, 3, 4, 2};
    int ttl_credits = 13;
    static String subjects[] = {"Sdm", "Oop", "dbms", "ds"};
    String usn;
    String name;
    static int n = 4;
    int marks_earned[] = new int[n];
    int marks_earned_gp[] = new int[4];
    int gp_c[] = new int[4];
    float tt_gp = 0;
    void put() {
```

```
Scanner Sx = new Scanner (System. In);
System.out.println ("enter the values of 81 : ");
System.out.print ("name : ");
name = Sx.nextLine();
System.out.println ("usn : ");
USN = Sx.nextLine();
System.out.println ("enter the marks : ");
for (int i = 0; i < stu.m; i++) {
    System.out.print (&stu.subjects[i] + " : ");
    marks Earned [i] = Sx.nextInt();
```

y
void display () {
 float GP = 0;
 System.out.println ("In In");
 System.out.println ("name : " + name + " | USN : " + USN);
 for (int i = 0; i < stu.m; i++) {
 if (marks Earned [i] >= 90 && marks Earned [i] < 90) {
 GP [i] = 9;
 }
 else if (marks Earned [i] >= 80 && marks Earned [i] < 90) {
 GP [i] = 8;
 }
 else if (marks Earned [i] >= 70 && marks Earned [i] < 80) {
 GP [i] = 7;
 }
 else if (marks Earned [i] >= 60 && marks Earned [i] < 70) {
 GP [i] = 6;
 }
 else if (marks Earned [i] >= 50 && marks Earned [i] < 60) {
 GP [i] = 5;

Else if (marks-earned[i] >= 40 & marks-earned[i]) {

 50) {

 gp[i] = 5;

 Else {

 gp[i] = 0;

 }

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

 gp-c[i] = g[i] * marks-credits[i];

 y

for (int i = 0; i < 4; i++) {

 y tt-gp = tt-gp + gp-c[i];

float sgpa = tt-gp / ttl-credits;

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

y

public class sgpa {

 public static void main (String args) {

 stu s1 = new stu();

 stu s2 = new stu();

 s1.show();

 s1.display();

y

O/P :

Enter number of subjects : 3.

Student name : vday

USN : 24 BE CS 403.

Enter max credits :

Sub 1 : 4

Sub 2 : 4

Sub 3 : 3.

Enter marks for each subject :

marks Sub 1 : 75

marks Sub 2 : 67

Multiple
student
object

marks for sub 3 & 7 9

Student details :

Name : Uday

U8N : 248ECS 403

SEM A : F.G.S.

3 : 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;
    String author;
    int price;
    int num_pages;

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

    book(){
        name = "";
        author = "";
        price = 0;
        num_pages = 0;
    }

    void set(){
        Scanner sx = new Scanner(System.in);
        System.out.print("enter the name of book : ");
        name = sx.next();
        System.out.print("enter the author of book : ");
        author = sx.next();
        System.out.print("enter the price of book : ");
        price = sx.nextInt();
        System.out.print("enter the number of pages of the book : ");
    }
}
```

```
    num_pages = sx.nextInt();

}

void get(){

    System.out.println(" name : "+name+"\n author : "+author+"\n price : "+price+"\n num of pages :
"+num_pages);

}

public String toString(){

    return " name : "+name+"\n author : "+author+"\n price : "+price+"\n num of pages : "+num_pages;

}

}

class lab3{

    public static void main (String args[]){

        int n;

        System.out.print("enter number of books : ");

        Scanner sx = new Scanner(System.in);

        n = sx.nextInt();

        book B[] = new book[n];

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

            System.out.println("enter details of book "+(i+1)+" : ");

            B[i] = new book();

            B[i].set();

            // B[i].get();

            System.out.println();

        }

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

        //     B[i].get();

        // }

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

            System.out.println("details of book "+(i+1)+" : ");

            System.out.println(B[i]);

        }

    }

}
```

```
}

book b1 = new book("rich dad poor dad","robert",250,150);

System.out.println();

System.out.println("parameterized constructor : ");

System.out.println(b1);

}

}

O/P :
```

```
PS D:\3rd sem\JAVA\new practice> javac lab3.java
PS D:\3rd sem\JAVA\new practice> java lab3
enter number of books : 2
enter details of book 1 :
enter the name of book : automic_habbits
enter the author of book : james_clear
enter the price of book : 280
enter the number of pages of the book : 145

enter details of book 2 :
enter the name of book : 1984
enter the author of book : george_orwell
enter the price of book : 300
enter the number of pages of the book : 153
```

```
details of book 1 :
name : automic_habbits
author : james_clear
price : 280
num of pages : 145
details of book 2 :
name : 1984
author : george_orwell
price : 300
num of pages : 153
```

```
parameterized constructor :
name : rich dad poor dad
author : robert
price : 250
num of pages : 150
```

3) 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 {

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 service (double price) £.

y this + price = price;

```
public void SetNumPages (int numPages) {
```

this.numPages = numPages;

public storing technique is

defeaten name:

public static getAuthor()

between author.

```
public static getAuthor ( ) {
```

stevens author

public double getprice()

return police;

public int getNunPages()

determine number of pages

④ override

public static void main() {

return "Book Details : 1n" &

"Name : " + name + " Ln " +

"author" "author + "in" + .

"price": \$ " + price + "ln" +

"Number of pages : " + numPages

3

Public class room of

public static void main (String [] args)

26

Scanner scanner < new scanner (System)

System auto printn ("enter number of Books")

```
int n = Scanner.nextInt();  
Scanner.nextLine();
```

Scanner - near Lincs. Eng.

```
Book[] books = new Book[n];
for (int i=0; i<n; i++) {
    System.out.println("Enter details for book " + i);
    System.out.print(" : ");
    String name = Scanner.nextLine();
    System.out.print(" Enter author name : ");
    String author = Scanner.nextLine();
    System.out.print(" Enter price : ");
    double price = Scanner.nextDouble();
    System.out.print(" Enter number of pages : ");
    int numPages = Scanner.nextInt();
    Scanner.nextLine();
    books[i] = new Book(name, author, price,
                        numPages);
```

y
y
y

```
System.out.println("In Details of all books");
for (int i=0; i<n; i++) {
    System.out.println(books[i].toString());
}
Scanner.close();
```

o/p

number of books : 1.

Enter details of Book 1 :

Enter book name : qwert

author name : zxcv

Enter price : 560

number of pages : 200

Details of all books :

Book details : ✓^{ab}

Name : QUERT

Author : ZICV.

price : \$560.0.

page : 200

4. 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.

```
import java.util.Scanner; abstract class Shape{  
    abstract public void printArea(); int i1,i2;  
}  
  
class Rectangle extends Shape{ public void printArea(){  
    Scanner sx = new Scanner(System.in); System.out.println("Area of  
    Rectangle"); System.out.print("enter length : ");  
    i1 = sx.nextInt();  
    System.out.print("enter breadth : "); i2 = sx.nextInt();  
    System.out.println("area of rectangle : "+(i1*i2)+"\n");  
}  
}  
  
class Triangle extends Shape{ public void printArea(){  
    Scanner sx = new Scanner(System.in); System.out.println("Area of  
    Triangle"); System.out.print("enter length : ");  
    i1 = sx.nextInt();  
    System.out.print("enter breadth : "); i2 = sx.nextInt();  
    System.out.println("area of Triangle : "+(0.5*i1*i2)+"\n");  
}  
}  
  
class Circle extends Shape{  
    public void printArea(){  
        Scanner sx = new Scanner(System.in); System.out.println("Area of  
        Circle"); System.out.print("enter radius : ");
```

```
i1 = sx.nextInt();

System.out.println("area of Circle : "+(3.14*i1*i1)+"\n");

}

}

class lab4{

    public static void main(String args[]){ Shape s = new Rectangle();

        s.printArea();

        s = new Triangle(); s.printArea();

        s = new Circle(); s.printArea();

    }

}
```

O/P:

```
Area of Rectangle
enter length : 3
enter bredth : 2
area of rectangle : 6

Area of Triangle
enter length : 3
enter bredth : 2
area of Triangle : 3.0

Area of Circle
enter radius : 3
area of Circle : 28.25999999999998
```

4) 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 the method printArea() that prints the area of the given shape. * /

→ Import java.util.Scanner
 abstract class Shape {
 abstract void printArea();
 }

class Triangle extends Shape {
 void printArea() {
 System.out.println ("The area of
 triangle");
 }
}

Scanner sc = new Scanner (System.in);
 int b, h;

double res;

System.out.println ("Enter base : ");
 b = sc.nextInt();

System.out.println ("Enter height : ");
 h = sc.nextInt();

res = (0.5 * b * h);

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

Y.	area of triangle b : 3 h : 4 area : 6.0.	area of rectangle b : 2 l : 3 area : 6.0	area of circle r : 4 area : 50.24
Y.			

class rectangle extends shape {
void paintArea () {
System.out.println ("In area of rectangle");
Scanner sc = new Scanner (System.in);
int b, l;
double res;
System.out.print ("Enter breadth : ");
b = sc.nextInt ();
System.out.print ("Enter length : ");
l = sc.nextInt ();
res = (b * l);
System.out.println ("Area of rectangle : " + res);
y
y

class circle extends shape {
void paintArea () {
System.out.println ("In area of circle");
Scanner sc = new Scanner (System.in);
int r;
double res;
System.out.print ("Enter radius : ");
r = sc.nextInt ();
res = (3.14 * r * r);
System.out.println ("Area of circle : " + res);
y
y.

class ab-pgm4 {
public static void main (String args[]) {
~~t~~ triangle t = new triangle ();
~~s~~ rectangle s = new rectangle ();
~~c~~ circle c = new circle ();
t.paintArea ();
s.paintArea ();
c.paintArea ();
y
y
y

area of triangle

$$b : 3$$

$$h : 4$$

$$\text{area} : 6.0.$$

area of rectangle

$$b : 2$$

$$l : 3$$

$$\text{area} : 6.0$$

area of circle

$$r : 4$$

$$\text{area} : 50.24$$

5. 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 Customer_name;
    Long account_number;
    String type;

    void getd() {
        Scanner sx = new Scanner(System.in);
        System.out.print("Enter Customer name: ");
        Customer_name = sx.next();
        System.out.print("Enter Customer Account number: ");
        account_number = sx.nextLong();
        System.out.print("Enter Account type (Savings/Current): ");
        type = sx.next();
    }
}

class Sav_acct extends Account {

    double balance;
    double interestRate = 0.05; // Annual interest rate

    void getad() {
        super.getd();
    }
}
```

```
}

void deposit() {
    Scanner sx = new Scanner(System.in);
    System.out.print("Enter Amount to deposit: ");
    double dep_amt = sx.nextDouble();
    balance += dep_amt;
    System.out.println("Your balance after depositing: " + balance);
}

void balance() {
    System.out.println("Your current balance: " + balance);
}

void compute_deposit_interest() {
    double interest = balance * interestRate;
    balance += interest;
    System.out.println("Interest added: " + interest);
    System.out.println("Your updated balance: " + balance);
}

void withdraw() {
    Scanner sx = new Scanner(System.in);
    System.out.print("Enter amount to withdraw: ");
    double wdrwl_amt = sx.nextDouble();
    if (wdrwl_amt > balance) {
        System.out.println("Insufficient balance!");
    } else {
        balance -= wdrwl_amt;
        System.out.println("Amount has been successfully withdrawn!");
        System.out.println("Your updated balance: " + balance);
    }
}

class Curr_acct extends Account {
```

```
double balance;

final double minimumBalance = 500.0;

final double penalty = 50.0;

void getad() {

    super.getd();

}

void deposit() {

    Scanner sx = new Scanner(System.in);

    System.out.print("Enter Amount to deposit: ");

    double dep_amt = sx.nextDouble();

    balance += dep_amt;

    System.out.println("Your balance after depositing: " + balance);

}

void balance() {

    System.out.println("Your current balance: " + balance);

}

void withdraw() {

    Scanner sx = new Scanner(System.in);

    System.out.print("Enter amount to withdraw: ");

    double wdrwl_amt = sx.nextDouble();

    if (wdrwl_amt > balance) {

        System.out.println("Insufficient balance!");

    } else {

        balance -= wdrwl_amt;

        if (balance < minimumBalance) {

            System.out.println("Your balance is below the minimum. Penalty imposed!");

            balance -= penalty;

        }

        System.out.println("Your updated balance: " + balance);

    }

}
```

```
}

class lab5 {

    public static void main(String args[]) {

        Scanner sx = new Scanner(System.in);

        Sav_acct sav = new Sav_acct();

        Curr_acct cur = new Curr_acct();

        while (true) {

            System.out.println("\nPress accordingly:");

            System.out.println("1. New Savings Account");

            System.out.println("2. New Current Account");

            System.out.println("3. Deposit");

            System.out.println("4. Check Balance");

            System.out.println("5. Withdraw");

            System.out.println("6. Calculate Interest (Savings only)");

            System.out.println("7. Exit");

            System.out.print("Enter your choice: ");

            int choice = sx.nextInt();

            switch (choice) {

                case 1:

                    sav.getad();

                    System.out.println("New Savings account has been created!");

                    break;

                case 2:

                    cur.getad();

                    System.out.println("New Current account has been created!");

                    break;

                case 3:

                    System.out.print("Deposit to (1: Savings, 2: Current): ");

                    int depChoice = sx.nextInt();

                    if (depChoice == 1) sav.deposit();
```

```
        else if (depChoice == 2) cur.deposit();
        else System.out.println("Invalid choice!");
        break;

    case 4:
        System.out.print("Check balance for (1: Savings, 2: Current): ");
        int balChoice = sx.nextInt();
        if (balChoice == 1) sav.balance();
        else if (balChoice == 2) cur.balance();
        else System.out.println("Invalid choice!");
        break;

    case 5:
        System.out.print("Withdraw from (1: Savings, 2: Current): ");
        int wdrChoice = sx.nextInt();
        if (wdrChoice == 1) sav.withdraw();
        else if (wdrChoice == 2) cur.withdraw();
        else System.out.println("Invalid choice!");
        break;

    case 6:
        sav.compute_deposit_interest();
        break;

    case 7:
        System.out.println("Exiting. Thank you!");
        return;

    default:
        System.out.println("Invalid choice. Please try again.");
    }
}
```

O/P :

Press accordingly:

1. New Savings Account
2. New Current Account
3. Deposit
4. Check Balance
5. Withdraw
6. Calculate Interest (Savings only)
7. Exit

Enter your choice: 1

Enter Customer name: uday

Enter Customer Account number: 7869574456

Enter Account type (Savings/Current): Savings

New Savings account has been created!

Press accordingly:

1. New Savings Account
2. New Current Account
3. Deposit
4. Check Balance
5. Withdraw
6. Calculate Interest (Savings only)
7. Exit

Enter your choice: 3

Deposit to (1: Savings, 2: Current): 1

Enter Amount to deposit: 25

Your balance after depositing: 25.0

Press accordingly:

1. New Savings Account
2. New Current Account
3. Deposit
4. Check Balance
5. Withdraw
6. Calculate Interest (Savings only)
7. Exit

Enter your choice: 5

Withdraw from (1: Savings, 2: Current): 1

Enter amount to withdraw: 26

Insufficient balance!

Press accordingly:

1. New Savings Account
2. New Current Account
3. Deposit
4. Check Balance
5. Withdraw
6. Calculate Interest (Savings only)
7. Exit

Enter your choice: 6

Interest added: 1.25

Your updated balance: 26.25

Press accordingly:

1. New Savings Account
2. New Current Account
3. Deposit
4. Check Balance
5. Withdraw
6. Calculate Interest (Savings only)
7. Exit

Enter your choice: 5

Withdraw from (1: Savings, 2: Current): 1

Enter amount to withdraw: 6

Amount has been successfully withdrawn!

Your updated balance: 20.25

Press accordingly:

1. New Savings Account
2. New Current Account
3. Deposit
4. Check Balance
5. Withdraw
6. Calculate Interest (Savings only)
7. Exit

Enter your choice: █

5) Develop a Java program to create a class Bank that maintains two kinds of account for the 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 requirement. Include the necessary methods in order to achieve the following tasks.

- a) Accept deposit from customer & 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;

long accountNumber;

double balance;

public void acceptDeposit(double amount) {
balance += amount;

```
public void display() {  
    System.out.println("Balance : "+balance);  
}  
  
public void getDetails() {  
    System.out.println("Customer Name "+customer  
        Name + " Account Number : "+account  
        number);  
}
```

class CurrentAccount extends Account {
 private static final double MIN-BALANCE
 = 2000;
 private static final double PENALTY = 500;

```
public void acceptDeposit(double amount) {  
    super.acceptDeposit(amount);  
    if (balance < MIN-BALANCE) {  
        System.out.println("Penalty  
imposed for not maintaining minimum  
balance.");  
        balance -= PENALTY;  
    }  
}
```

class SavingAccount extends Account {
 private static final double INTREST-
 RATE = 0.05;

```
public void computeAndDepositInterest(int years) {  
    double compoundInterest = balance * Math.  
        pow(1 + INTREST RATE, years) -  
        balance;
```

balance += compoundInterest;

}
public void withdraw (double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println (" withdrawal successful");

Amount withdrawn : " + amount);

}
else {

System.out.println (" insufficient balance
withdrawal unsuccessful . ");

public class Lab5 {

public static void main (String args) {

Scanner scanner = new Scanner (System.in);

CurrentAccount currentAccount = new Current

SavingsAccount savingsAccount = new Savings
Account();

System.out.println (" Enter customer name : ");

currentAccount.customerName = scanner.nextLine();

CustomerName = scanner.nextLine();

System.out.println (" Enter account number : ");

CurrentAccount.accountNumber = scanner.nextLine();

accountNumber = scanner.nextLine();

Ent choice ;

do {

```
System.out.println ("1. menu");  
System.out.println ("1. Deposit to current  
Account");  
System.out.println ("2. Deposit to Savings  
Account");  
System.out.println ("3. Display Balance for  
current Account");  
System.out.println ("4. Display Balance for  
Savings Account");  
System.out.println ("5. Compute Interest for  
Savings Account");  
System.out.println ("6. withdraw from Savings  
Account");  
System.out.println ("7. Exit");  
System.out.println ("Enter your choice");  
choice = Scanner.nextInt();
```

~~switch (choice) {~~

~~case 1:~~

```
S.O.P ("Enter amount to deposit to,  
current Account : ");  
double currentDeposit = Scanner.nextDouble();  
CurrentAccount.acceptDeposit (currentDeposit);  
break;
```

~~case 2:~~

```
System.out.println ("Enter amount to  
deposit to Savings Account : ");  
double savingsDeposit = Scanner.nextDouble();  
SavingsAccount.acceptDeposit (savingsDeposit);  
break;
```

Code 3 :

```
s. o. p ("Current Account Balance : ");
currentAccount. displayBalance ();
break;
```

Code 4 :

```
s. o. p ("Savings Account Balance : ");
savingsAccount. displayBalance ();
break;
```

Code 5 :

```
System. out. print ("Enter the years for
interest calculation : ");
int years = Scanner. nextInt ();
```

```
Savings Account. computeAndDeposit (years);
```

```
s. o. p ("Interest computed & deposited ");
break;
```

Code 6 :

```
s. o. p ("Enter amount to withdraw, from
Savings Account : ");
double withdrawAmount = Scanner. nextDouble ();
```

```
Savings Account. withdraw (withdrawAmount);
break;
```

Code 7 :

```
s. o. p ("Exiting program ... ");
break;
```

default :

```
s. o. p ("Invalid choice. Try again ");
y
```

```
y while (choice != 7);
y
```

```
y Scanner. close ();
```

```
y
```

O/P : Enter customer name : voley
Enter account number : 56457456

Menu :

- 1 - Deposit to current account
- 2 - Deposit to Savings Account
- 3 - Display Balance for current account
- 4 - Display Balance for Savings Account
- 5 - Compute Interest for Savings Account
- 6 - withdraw from Savings Account
- 7 - Exit

Enter your choice :

Enter amount to deposit to current account ? 5000

Menu :

Enter your choice : 2.

Enter amount to deposit to Savings Account ? 3000

Menu :

Enter your choice : 3.

Current Account Balance :

Balance : 5000.0

Menu :

Enter your choice : 4

Savings Account Balance :

Balance : 3000.0

Menu :

Enter your choice : 5

Enter the number of years for interest

Interest calculated : 5
Interest computed and deposited

Menu :

choice : 4

Savings Account Balance :

Balance : 3828. \$ 4468.

Menu :

choice * 6.

enter Amount to withdraw : 300

Successfully withdraw : 300

Menu :

choice * 4.

Balance : 3528. \$ 847.17

6. Create a 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 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;

public class Personal {
    protected String usn;
    protected String name;
    protected int sem;

    public Personal(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }

    public String getUsn() {
        return usn;
    }

    public String getName() {
        return name;
    }

    public int getSem() {
        return sem;
    }
}

// CIE/Internals.java
package CIE;

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

    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }

    public int[] getInternalMarks() {
        return internalMarks;
    }
}
```

```
}
```

// SEE/External.java

```
package SEE;

import CIE.Personal;

public class External extends Personal {
    private int[] seeMarks = new int[5];

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

    public int[] getSeeMarks() {
        return seeMarks;
    }
}
```

// Main.java

```
import CIE.*;
import SEE.*;

import java.util.Scanner;

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

        System.out.print("Enter the number of students: ");
        int n = scanner.nextInt();

        Personal[] students = new Personal[n];

        // Input for students in CIE and SEE
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for student " + (i + 1));

            // Input for Personal Information
            System.out.print("USN: ");
            String usn = scanner.next();
            System.out.print("Name: ");
            String name = scanner.next();
            System.out.print("Semester: ");
            int sem = scanner.nextInt();

            // Input for Internal marks (CIE)
```

```

int[] internalMarks = new int[5];
System.out.println("Enter internal marks for 5 courses:");
for (int j = 0; j < 5; j++) {
    internalMarks[j] = scanner.nextInt();
}

// Create Internals object
Internals internals = new Internals(usn, name, sem, internalMarks);

// Input for External marks (SEE)
int[] seeMarks = new int[5];
System.out.println("Enter external marks for 5 courses:");
for (int j = 0; j < 5; j++) {
    seeMarks[j] = scanner.nextInt();
}

// Create External object
External external = new External(usn, name, sem, seeMarks);

// Calculate and display the final marks
System.out.println("\nStudent Details:");
System.out.println("USN: " + internals.getUsn());
System.out.println("Name: " + internals.getName());
System.out.println("Semester: " + internals.getSem());

System.out.println("\nInternal Marks:");
int[] internalMarksArr = internals.getInternalMarks();
for (int j = 0; j < 5; j++) {
    System.out.print(internalMarksArr[j] + " ");
}

System.out.println("\nExternal Marks:");
int[] seeMarksArr = external.getSeeMarks();
for (int j = 0; j < 5; j++) {
    System.out.print(seeMarksArr[j] + " ");
}

// Calculate final marks
int totalMarks = 0;
for (int j = 0; j < 5; j++) {
    totalMarks += internalMarksArr[j] + seeMarksArr[j];
}

System.out.println("\nFinal Marks (Total): " + totalMarks);
}

scanner.close();

```

```
    }  
}
```

O/P :

```
Enter the number of students: 1  
  
Enter details for student 1  
USN: 1BM21CS001  
Name: John  
Semester: 5  
Enter internal marks for 5 courses:  
20 18 22 25 20  
Enter external marks for 5 courses:  
40 45 35 50 40  
  
Student Details:  
USN: 1BM21CS001  
Name: John  
Semester: 5  
  
Internal Marks:  
20 18 22 25 20  
External Marks:  
40 45 35 50 40  
Final Marks (Total): 320
```

6. Create a 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 External which is a derived class of student. This class has an array that stores the SBE 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;

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;

class. sum = sum ;

3
public void displayDetails() {

System.out.println("User : " + user);

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

System.out.println("Gender : " + gender);

3

3

Internals.java

package CIB;

public class Internals {

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

public Internals (int[] marks) {

if (marks.length == 5).

System.arraycopy(marks, 0, Internals

marks, 0, marks.length);

3

else {

System.out.println("Error : Exactly
5 external marks are required.");

3

public void displayInternalMarks() {

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

for (int mark : internalMarks) {

System.out.print(mark + " ");

System.out.println();

External.java

```
package SEE;
import CIE.Student;

public class External extends Student {
    public int[] ExternalMarks = new int[5];
    public External (String usn, String name,
                    int sem, int[] marks) {
        super (usn, name, sem);
        if (marks.length == 5)
    }
}
```

```
System.arraycopy (marks, 0, externalMarks,
                  0, marks.length);
```

```
else {
```

```
System.out.println ("Error : Exactly  
5 SEE marks are required.");
```

```
public void displayExternalMarks () {
    System.out.print ("SEE marks : ");
    for (int mark : externalMarks) {
        System.out.print (mark + " ");
    }
    System.out.println ();
```

FinalMarks.java

```
import CIE.*;
import SEE.*;
import java.util.Scanner;
```

```

public class FinalMarks {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of Students:");
        int n = sc.nextInt();
        Student[] students = new Student[n];
        Internal[] internals = new Internal[n];
        External[] externals = new External[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter details of Student " + (i + 1) + ":");
            System.out.print("USN: ");
            String usn = sc.next();
            System.out.print("Name: ");
            String name = sc.next();
            System.out.print("Semester: ");
            int sem = sc.nextInt();
            System.out.println("Enter 5 Internal marks: ");
            int[] internalMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                internalMarks[j] = sc.nextInt();
            }
            System.out.println("Enter 5 SEE marks: ");
            int[] externalMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                externalMarks[j] = sc.nextInt();
            }
            students[i] = new Student(usn, name, sem);
            internals[i] = new Internal();
            externals[i] = new External();
        }
    }
}

```

internalmarks);
externals[i] = new External(vsn, name, sem, external
marks);

3)

System.out.println("Final marks of Student");

```
for (int i=0; i<n; i++) {  
    students[i].displayDetails();  
    internals[i].displayInternalMarks();  
    external[i].displayExternalMarks();
```

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

```
for (int j=0; j<5; j++) {  
    int finalmarks = internals[i].internal  
    externals[i].externalmarks[j].marks[j]/2;
```

System.out.print(finalmarks + " ");

System.out.println("in ");

3) ~~Final marks = internalmarks + externalmarks / 2;~~

Q) O/P. Enter no of Students : 1
Enter Student details.

Enter UBN : 24BEC8403

Enter Name : Uday

Enter Sem : 5

Enter CIE marks : 20 12 15 22 19

Enter 5 See marks : 60 70 65 55 75

Student marks details

UBN : 24BEC8403.

Name : Uday.

Sem : 5

Final marks :

50 53 58 69 56

7. Write a program that demonstrates handling of exceptions in inheritance tree.
Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age=father’s age.

```
import java.util.Scanner;

class WrongAge extends Exception{

    public WrongAge(String message){

        super(message);

    }
}

class father{

    int age;

    father(){

        Scanner sx = new Scanner(System.in);

        System.out.print("enter father age : ");

        age = sx.nextInt();




        try{
            if(age<0){

                throw new WrongAge("age cannot be negative...!");

            }
        }

        catch(WrongAge we){

            System.out.print(we);

        }
    }
}

class son extends father{



    son(){

        super();

        int son_age;
    }
}
```

```

Scanner sx = new Scanner(System.in);
System.out.print("enter son age : ");
son_age = sx.nextInt();
try{
    if(son_age > age){
        throw new WrongAge("son age cannot be greater than father age...!");
    }
}
catch(WrongAge we){
    System.out.print(we);
}
}

public class lab7{
    public static void main(String args[]){
        // father fa = new father();
        son sa = new son();
    }
}

```

O/P:

```

enter father age : 45
enter son age : 50
WrongAge: son age cannot be greater than father age...!
PS D:\3rd sem\JAVA\new practice> java lab7
enter father age : -2
WrongAge: age cannot be negative...!enter son age : 4
WrongAge: son age cannot be greater than father age...!
PS D:\3rd sem\JAVA\new practice> java lab7
enter father age : -4
WrongAge: age cannot be negative...!enter son age : 3
WrongAge: son age cannot be greater than father age...!

```

7) Write a program that demonstrates handling of exception inheritance. Create a base class called 'Father' and derived class called Son which extends the base class. In Father class; implement a constructor which takes the age and throws the exception WrongAge() when the input age < 0 in Son's class. Implement a constructor that uses both father and son's age and throw an exception if son's age >= father's age.

→ class WrongAge extends Exception {
 WrongAge (String message) {
 super (message);
 }
}

3.
class Father {
 int age;
 Father (int age) {
 if (age < 0) {
 throw new WrongAge ("age cannot be negative");
 }
 this.age = age;
 }
}

3.
class Son extends Father {
 Son (int age, int fatherAge) {
 super (age, fatherAge);
 if (age >= fatherAge) {
 throw WrongAge();
 }
 }
}

+ how new ~~exception~~ wrong Age ("
Son age can't greater than father age");

y

this. fage = fage ;

this. sage = sage ;

y

class reader {

perm C wrongAge(); {

for(;;) {

Scanner sc = new Scanner (System.in);

try {

int fage;

int sage;

System.out.print ("Enter father age ");

fage = sc.nextInt();

System.out.print ("Enter son age: ");

sage = sc.nextInt();

Son s = new Son (fage, sage);

y

catch (wrongAge wa) {

System.out.println (wa);

op : Enter father's age :- 1.

Enter son's age = 1.

Error : Age cannot be negative.

Enter father's age :- 20

Enter son's age : 21

Error! : Son's age > father's age

Exception which creates two threads

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

```
class thread1 extends Thread{  
    public void run(){  
        while(true){  
            System.out.println("BMS College of Engineering");  
            try{  
                Thread.sleep(10000);  
            }  
            catch(InterruptedException e){  
                System.out.println(e);  
            }  
        }  
    }  
  
    class thread2 extends Thread{  
        public void run(){  
            while(true){  
                System.out.println("CSE");  
                try{  
                    Thread.sleep(2000);  
                }  
                catch(InterruptedException e){  
                    System.out.println(e);  
                }  
            }  
        }  
    }  
  
    class lab8{  
        public static void main(String args[]){
```

```
    thread1 t1 = new thread1();
    thread2 t2 = new thread2();
    t1.start();
    t2.start();
}
}
```

O/P :

```
PS D:\3rd sem\JAVA\new practice> java lab8
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
```

8. write a program which creates two threads
thread displaying BMS college of Engineering
every ten seconds and another display "CSE"
once every two seconds

→

```
class A extends Thread {  
    public void run() {  
        while (true) {  
            +sy {  
                System.out.println ("BMS College  
of Engineering");  
            thread . sleep (10000);  
        }  
    }  
}  
catch ( InterruptedException c ) {  
    System.out.println (c);  
}
```

y
y
y
y

```
class B extends Thread {  
    public void run() {  
        while (true) {  
            +sy {  
                System.out.println ("CSE");  
            thread . sleep (2000);  
        }  
    }  
}
```

y

catch (InterruptedException e)
System.out.println (e);

y

y

y

y
class threads_pgms

public static void main (String args[]) {

A obj1 = new A();

B obj2 = new B();

obj1.start();

obj2.start();

y

y

O/P : BM S college of Engineering

CSE

CSE

CSE

CSE

CSE

BM S college of Enggery.

~~25/12/2012~~