

Observations

1) Student Program

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

```
class Student
```

```
{ String name, usn;
```

```
int marks[ ] = new int[10];
```

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

```
void accept Details()
```

```
{ System.out.println ("Enter the USN");
```

```
USN = sn.nextInt();
```

```
System.out.println ("Enter the name");
```

```
name = sn.next();
```

```
System.out.println ("Enter 6 marks");
```

```
for (int i=0; i<6; i++).
```

```
{ marks[i] = sn.nextInt();
```

```
}
```

```
double calculate ()
```

```
{ int sum=0;
```

```
double per;
```

```
for (int i=0; i<6; i++)
```

```
{ sum = sum + marks[i]; }
```

```
per = sum / 6.0;
```

```
return per;
```

```
}
```

void display ()

{ System.out.println ("The name of the student is :
" + name); }

System.out.println ("The v/s% of the student is : " + v/s%);

System.out.println ("The percentage of student is : "
+ calculate () + "%"); };

public static void main (String args [])

{ Scanner sn = new Scanner (System.in); }

System.out.println ("Enter no. of students");

int stu = sn.nextInt();

Student [] student = new Student [stu];

for (int i = 0; i < stu; i++)

{ Student [i] = new Student (); }

System.out.println ("Enter the details of student");

student [i].setDetails ();

}

System.out.println ("Student details are : ");

for (int i = 0; i < stu; i++)

{ System.out.println ("Details of students " + (i + 1)
+ " are "); }

student [i].calculate ();

}

sn.close ();

}

O/P

Enter no. of students = 2

Enter details of student 1 :

Enter the name : Arush

Enter the USN : 1BM22CS000

Enter 6 marks : 45

66

90

85

71

99

Enter details of student 2 :

Enter name : Aditya.

Enter USN : 1BM22CS001

Enter 6 marks : 48

60

98

100

99

Student Details are :-

Name of student 1 is Arush

USN of student 1 is 1BM22CS000

The Percentage of student is 85.035%.

Name of student 2 is Aditya.

USN of student is 1BM22CS001

The percentage of student is 88.240%.

Quadratic Egn :

```
import java.util.*;
import java.lang.Math;
class Quad {
    double a, b, c, rd, r1, r2;
    void input() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the 1st coeffcient");
        a = sc.nextInt();
        System.out.println("Enter the 2nd coefficient");
        b = sc.nextInt();
        System.out.println("Enter the 3rd coefficient");
        c = sc.nextInt();
    }
    void calc() {
        double d = b * b - 4 * a * c;
        if (d > 0) {
            r1 = (b + Math.sqrt(d)) / (2 * a);
            r2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.println("The roots are real & distinct = " + r1 + " and " + r2);
        } else if (d < 0) {
            r = -b / (2 * a);
            t = Math.sqrt(-d) / (2 * a);
            System.out.println("The roots are real and equal : " + r);
        }
    }
}
```

System.out.println("The roots are ~~real~~ imaginary of
distinct : " + r1 + " + " + r2 + "i" and " + r + ". " + c + "i");

else {

r2 = b / (2 * a);

System.out.println("The roots are real and
equal : " + r);

y3

Class Quad {
public static void main (String args []) {
Quad q = new Quad ();
q . input ();
q . calc ();
y3

Enter the first coefficient : 1

Enter the second : 2

Enter the third : 1

Roots are real and equal : -1.0

Q8. Abstract class Shape {
 protected int side1;
 protected int side2;
 public shape (int side1, int side2) {
 this.side1 = side1;
 this.side2 = side2; }
 public abstract void printArea(); }

Class Rectangle extends Shape {
 public Rectangle (int length, int width) {
 super (length, width); } }

@ override

 public void printArea () {
 int area = side1 * side2;
 System.out.println ("Area of rectangle : " +
 area); }

Class Triangle extends Shape {

 public Triangle (int base, int height) {
 super (base, height); }

@ override

 public void printArea () {
 double area = 0.5 * side1 * side2;
 System.out.println ("Area of Triangle : " +
 area); }

Class Circle extends Shape {

 public Circle (int radius) {
 super (radius, 0); }

@ override

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

```
    double area = Math.PI * side1 * side2;
```

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

```
} }
```

```
Public class Main {
```

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

```
    Rectangle rectangle = new Rectangle (5, 10);
```

```
    rectangle.printArea();
```

```
    Triangle triangle = new Triangle (8, 6);
```

```
    triangle.printArea();
```

```
    Circle circle = new Circle (4);
```

```
    circle.printArea(); } }
```

~~Q.P~~

~~Area of Rectangle = 50~~

~~Area of Triangle = 24~~

~~Area of Circle = 50.264.~~

2) Bank account class :-

```
import java.util.Scanner;  
class Account {  
    boolean cheque;  
    String name;  
    long accno;  
    String acctype;  
    double bal;  
    Scanner sc = new Scanner (System.in);  
    Account (String name, long accno, String acctype,  
            double bal, boolean cheque) {  
        this.name = name;  
        this.accno = accno;  
        this.acctype = acctype;  
        this.bal = bal;  
        this.cheque = cheque; }  
    void deposit () {  
        System.out.println ("Enter amount to be  
                           deposited");  
        double amt = sn.nextDouble ();  
        bal += amt; }  
    void displayBal () {  
        System.out.println ("Balance: " + bal); }  
    public void interest () {  
        System.out.println ("Enter rate: ");  
        double rate = sn.nextDouble ();  
        if ("Savings".equals (acctype)) {  
            double interest = bal * (rate / 100);  
            bal += interest; } }
```

```
public void withdrawl(){
    System.out.println("Enter the amount to be
                        withdrawn");
    double amt = sc.nextInt();
    if(amt <= bal){
        bal -= amt;
    } else {
        System.out.println("Insufficient funds");
    }
}

class Savacct extends Account{
    public Savacct(String name, long accno,
                   double bal, boolean cheque){
        super(name, accno, "Savings", bal, cheque);
    }
}

class Currentacct extends Account{
    double minbal;
    double sav;
    public Current(String name, long accno, double
                   bal, double minbal, double sav, boolean
                   cheque){
        super(name, accno, "Current", bal, cheque);
        this.minbal = minbal;
        this.sav = sav;
    }
}
```

@ override

```
public void withdraw(){
    System.out.println("Enter amount to be
                        withdrawn");
    double amt = sc.nextInt();
    if(amt <= bal - minbal){
        bal -= amt;
    }
}
```

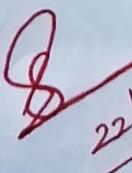
```
else {
    System.out.println("Insufficient funds"); }

public void check() {
    if (bal < minbal) {
        bal -= 50;
        System.out.println("Service charges - applied"); }
}

class Main {
    public static void main (String args[]) {
        SavingsAcct s = new SavingsAcct ("Joseph J.", "1234560",
                                         1000.0, false);
        CurrentAcct c = new Currentacct ("John Rila",
                                         "988695", "2500,00", "500", true);
        System.out.println ("Savings Acc : ");
        s.displayBal();
        s.dep();
        s.displayBal();
        s.interest();
        s.displayBal();
        s.withdraw();
        s.displayBal();
        System.out.println ("Currentacct : ");
        c.displayBal();
        c.dep();
        c.displayBal();
        c.withdraw();
        c.displayBal();
        c.check();
        c.displayBal(); }
```

~~OP :~~ Saving Acc :
Balance : 1000.0
Enter amount to be deposited : 250
Balance : 1200.0
Enter rate : 8
Balance : 1296.0
Enter amount to be withdrawn : 670
Balance : 626.0

Current Acc :
Balance : 2000.0
Amount to be deposited : 0
Balance : 2000.0
Amount to be withdrawn : 1600
Balance : 400
Service charge approved
Balance : 35


22/1/22

Package:

```
package CIE;
public class Student {
    public String usn, name;
    public int sem;
    public Student (String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem; }}
```

```
package CIE;
public class Internals extends Student {
    public double [] internalmarks = new double [5];
    public Internals (String usn, String name, int sem,
                      double [], internalmarks) {
        super (usn, name, sem);
        this.internalmarks = internalmarks; }}
```

```
package SEE;
import CIE.Student;
public class External extends Student {
    public double [] SeeMarks = new double [5];
    public External (String usn, String name, int sem,
                     double [] seeMarks) {
        super (usn, name, sem);
        this.seeMarks = seeMarks; }}
```

```
Package result;
import CIE.Student;
import CIE.Internals;
import SEE.External;
```

```

public class result {
    public static void main (String args [ ] ) {
        double internal [ ] = { 43, 45, 47, 49, 41 };
        double external [ ] = { 90, 87, 65, 98, 43 };
        Student s1 = new Student ("1BM 22CS150", "Jose");
        Internal i1 = new Internal ("1BM 22CS150", "Jose", "internal");
        External e1 = new External ("1BM 22CS150", "Jose", 3, "externals");
        System.out.println ("USN" + s1.usn + "name :" + s1.name
                            + "sem :" + s1.sem);
        S.O.P ("Internal marks");
        for (int i = 0; i < 5; i++) {
            S.O.P ("Internal marks" + (i + 1) + ":" + i1.internal
                   marks [i]);
        }
        S.O.P ("External Marks");
        for (int i = 0; i < 5; i++) {
            S.O.P ("External Marks :" + (i + 1) + ":" + e1.external
                   marks [i]);
        }
    }
}

```

OP USN : 1BM22CS0150 · name : Jose sem : 3

Internal Marks	1 :	43
	2 :	45
	3 :	47
	4 :	44
	5 :	41

External Marks	1 :	90
	2 :	87
	3 :	65
	4 :	98
	5 :	43

Father - son exception :

class WrongAge extends Exception {
public WrongAge() {
super ("Age cannot be negative"); } }

class Father {

private int age ;

public Father (int age) throws WrongAge {
if (age < 0) {

throw new WrongAge(); } }

this . age = age ;

public int getAge () {

return age; } }

class Son extends Father {

private int sonAge ;

public Son (int fatherAge , int sonAge) throws
WrongAge , IllegalArgumentException {

super (FatherAge);

if (sonAge >= FatherAge)

throw new IllegalArgumentException (
"Son's age must be less than Father's")

this . sonAge = sonAge ;

public int getSonAge () {

return sonAge; } }

public class - Exception Handling Inheritance {

public static void main (String args []) {

try {

Father . Father = new Father (age : 50);

Son . son = new Son (Father . Age : 50, Son . Age : 25);
System . out . println ("Father's age : " + Father .

System . out . println (Son . getAge ()); } }

System . out . println ("Son's age : " + son . getSon . age()); } }

```
catch (WrongAge e) {
```

```
    System.out.println("Exception :" + e.getMessage());
```

```
catch (IllegalArgument Exception e) {
```

```
    System.out.println("Exception :" + e.getMessage());
```

```
}
```

Book Program

```
import java.util.*;  
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 String getName(){  
        return name;  
    }  
    public String getAuthor(){  
        return author;  
    }  
    public double getPrice(){  
        return price;  
    }  
    public int getNumPages(){  
        return 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;  
    }
```

@ override

public String toString(){

return "Book Details : \n Name : " + name + "\n + " + author + "\n Price : " + price + "\n .

Number of pages : " + num - pages; } }

public class BookTest{

public static void main (String args[]){

Scanner sn = new Scanner (System.in);

System.out.println ("Enter no. of books");

int n = sn.nextInt();

Book [] book = new Book[n];

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

System.out.println ("Enter details for book " +

(i+1) + ": ");

System.out.print ("Name ");

String name = sn.nextLine();

System.out.print ("Author ");

String author = sn.nextLine();

System.out.print ("Price ");

double price = sn.nextDouble();

System.out.print ("Number of pages : ");

int num - pages = sn.nextInt();

books[i] = new Book(name, author, price, num -
pages); }

System.out.println ("Details of all books : ");

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

System.out.println ("Book " + (i+1) + ": ");

System.out.println (books[i]); }

sn.close(); }

Enter number of books : 2

Enter details of Book 1

Name : Harry Potter

Author : J.K. Rowling

Price : 400

Number of pages : 300

Enter details of Book 2

Name : The Catcher in the Rye

Author : J.D. Salinger

Price : 380

Number of pages : 328

Details of all books :

Book 1 :

Name : Harry Potter

Author : J.K. Rowling

Price : 400 \$

No. of pages : 300

Book 2 :

Name : The catcher in the Rye

Author : J.D. Salinger

Price : \$ 380

No. of pages : 328

Thread Program

class Display Message extends Thread {

private String message;

private int interval;

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

@ override

public void run () {

while (true) {

System.out.println (message);

try {

Thread.sleep (interval * 1000); }
catch (InterruptedException e) {

e.printStackTrace (); }
}}}

public class Main {

public static void main (String args []) {

Display Message thread 1 = new Display.Message

(message : "BMS College of engineering",
interval : 10);

Display Message thread 2 = new Display.Message

(message : "CSE", interval : 2);

thread 1.start ();

thread 2.start (); } }

O/P

BMS college of Engineering

BSE

BMS college of Engineering

CSE

CSE

BMS college of Engineering

CSE

CSE

CSE

BMS college of Engineering

CSE

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...