Quadratic Equation:

```
import java.lang.Math.*;
import java.util.Scanner;

class Quadratic {
    double a, b, c, firstroot, secondroot;

    Quadratic (double a, double b, double c)
    {
      this.a = a;
      this.b = b;
      this.c = c;
    }
  void Eval()
    {
      firstroot = (-b + Math.sqrt(det))/(2*a);
      secondroot = (-b - Math.sqrt(det))/(2*a);
      System.out.format("First root = %.2f and
                         Second root = %.2f",
                         firstroot, secondroot);
    }
  else if (det==0)
  {
    firstroot = secondroot = -b/(2*a);
    S.O.F ("First Root = Second Root = %.2f ;",
            firstroot);
  }
```

```
else {
double real = -b/(2*a);
double img = Math.sqrt(-det)/(2*a);
System.out.printf("first root = %.2f +%.2fi\n",
                  real, img);
S.O.P ("Second Root = %.2f - %.2fi", real,
                  img);
}
}
}
Class Quad {
    public static void main(String args[])
    {
        double a, b, c, firstroot, secondroot;
        Scanner sc = new Scanner(System.in);
        S.O.P ("Enter a :");
        a = sc.nextdouble();
        S.O.P ("Enter b : ");
        b = sc.nextDouble();
        S.O.P ("Enter c : ");
        c = sc.nextDouble();

        Quadratic q = new Quadratic (a, b, c);
        q.Eval();
    }
}
```

~~Student program~~

```
import java.util.Scanner;

class Student {
    String[] usn;
    String[] name;
double ~~String~~[][] marks;

  Student (int nos, int noStudents)
  {
   usn = new String [number of students];
   name = new String [number of students];
   marks = new double [number of students]
                      [number of subjects];
  }

void accept Details (int studentIndex)
   {
     Scanner s = new Scanner (System.in);
     SOP ("Enter USN for student");
     usn [studentIndex] = ~~Sc~~.nextLine();
     SOP ("Enter ~~for~~ marks");
     for (int i=0; i<marks [studentIndex].length;
                                    i++)
     {
       System.out.print ("subject");
       marks[studentIndex] [i] = s.nextDouble();
     }
}
```

```
double calcPercentage (int studentIndex)
   {
     double totalMarks = 0;
     for (double mark : marks [studentIndex])
     {
        totalMarks += mark;
     }
    return (totalMarks / marks [studentIndex].length);
  }


void displayDetails (int studentIndex).
{
   SOP (" Student" + i);
   SOP ("USN:" + usn[sIndex]);
   SOP ("Name:" + name [sIndex]);
   ~~Sc~~ SOP ("Marks:");

   for (int i=0; i<marks [studentIndex], length; i++)
{
   SOP ("subject" + (i+1) + ":" + marks[studentIndex]
}
   SOP ("Percentage:" + ---);
  }
}
```

```
public class Main {
    psvm (String args[])
    {
      Scanner s = new Scanner(System.in);
      System.out.print("Enter, Students : ");
      int noOfStudents = s.nextInt();
    Student StudentObj = new student[noOfStudents];
    for (int i=0; i< noOf students ; i++)
    {
      StudentObj. acceptDetails(i);
    }
    for(int i= 0; i< noOf students ; i++)
    {
      StudentObj. displayDetails(i);
    }
  }
}
```

Output :

```
Enter USN: 1BM22CS096
Enter Name: Mithun G
Sub 1: 90
Sub 2: 82
Sub 3: 86
Sub 4: 90
Sub 5: 88
Sub 6: 86
Total percentage : 89
```

Book DataBase :

```
import java.util.Scanner;

class Books
{
  String name;
  String author;
  int price;
  int numPages;

  Books() {};
  Books(String name, String author, int price, int numPages);
  {
    this.name = name;
    this.author = author;
    this.price = price;
    this.numPages = numPages;
  }

public String toString() {
    String name, author, price, numPages;
    name = "Book name: " + this.name + "\n";
    author = "Author name: " + this.author + "\n";
    price = "Price: " + this.price + "\n";
    numPages = "Number of pages" + this.numPages + "\n";

    return name + author + price + numPages;
  }
}
```

```
class Bmain {
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int n;
        String name, author;
        int price, numPages;
        String

        System.out.println("Enter number of books:");
        n=sc.nextInt();

        Books b[];
        b= new Books[n];

        for(int i=0;i<n;i++) {
            S.o.p("Books"+(i+1)+":");
            S.o.p("Enter book:");
            name = sc.next();
            S.o.p("Enter Author:");
            author = sc.next();
            sop("Enter price:");
            price = sc.nextInt();
            sop("Enter no of pages:");
            numPages = sc.nextInt();
            b[i] = new Books(name, author, price, numPages);
        }
        for(int i=0;i<n;i++) {
            S.o.p("Book:"+(i+1)+"\n"+b[i]);
        }
    }
}
```

Output:

Enter no of books → 1

Books-1:

Enter name: Ramayana

Enter Author: Valmiki

Enter price: 256 Rs

Enter number of pages: 672 pgs

Book 1:

Book name: Ramayana

Author name: Valmiki

Price: 256 Rs

Number of Pages: 672 pgs

Program 2:
Abstract Method / Class

```
abstract class Shape
    {
        public int side1, side2;
        abstract void printArea();
    }

class Rectangle extends Shape
    {
        Rectangle(int length, int breadth) {
            this.side1 = length;
            this.side2 = breadth;
        }
    void printArea() {
        S.o.p("Area of rectangle "+(side1*side2));
        }
    }

class Triangle extends Shape {
        Triangle(int base, int height) {
            this.side1 = base;
            this.side2 = height;
        }
    void printArea() {
        S.o.p("The area of triangle "+(0.5
                *side1*side2));
        }
```

```
class Circle extends Shape {
        Circle(int rad) {
            this.side1 = this.side2 = rad;
        }
    void printArea() {
        S.o.p("The area of circle "+(side1*side2
                *3.14));
        }
}

class BRun {
    public static void main(String[] args) {
        Rectangle r = new Rectangle(10,10);
        Triangle t = new Triangle(5,10);
        Circle c = new Circle(5);
        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

Output:
Area of Rectangle: 100
Area of Triangle: 25.0
Area of Circle: 78.5

Program 10:

Bank Class

```
import java.util.Scanner;

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

    Account(String customerName, int accountNumber,
        String accountType, double balance)
    {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }
    abstract void deposit(double amount);
    abstract void displayBalance();
    abstract void computeInterest();
    abstract void withdraw(double amount);
}

class SavingsAccount extends Account {
    SavingsAccount(String customerName, int accountNumber
            String accountType, double balance)
        {
            super(custname, balance, accNumber, accType);
        }
    void deposit(double amount) {
        balance += amount;
        S.o.p("Amount deposited:" + amount);
    }
    void displayBalance() {
        S.o.p("Balance:" + balance);
    }
    void computeInterest()
    {
        double interestRate = 0.05;
        double interest = balance * interestRate;
        balance += interest;
        S.o.p("Interest Added" + interest);
    }
    void withdraw(double amount) {
        if (balance < amount)
        {
            S.o.p("Insufficient balance");
        }
        else
        {
            balance -= amount;
            S.o.p("Amount withdrawn:" + amount);
        }
    }
}
```

```
class CurrentAccount extends Account {
    double serviceCharge = 80;
CurrentAccount(String customerName, int accountNumber,
    String accountType, double balance)
{
    super(customerName, accountNumber, accountType,
        balance);
}
void deposit(double amount)
{
    balance += amount;
    System.out.println("Amount deposited: " + amount);
}
void displayBalance() {
    System.out.println("Balance: " + balance);
}
void computeInterest() {
    System.out.println("Current account does not have interest");
}
void withdraw(double amount) {
    if(balance - amount < 1000)
    {
        System.out.println("Insufficient balance");
        balance -= serviceCharge;
        System.out.println("ServiceCharges: " + serviceCharge);
    }
    else {
        balance -= amount;
        System.out.println("Amount withdrawn: " + amount);
```

```
class BRun {
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter customer name : ");
        String customerName = sc.nextLine();
        System.out.print("Enter account number : ");
        int accountNumber = sc.nextInt();
        System.out.print("Enter account type : ");
        String accountType = sc.next();
        System.out.print("Enter initial balance : ");
        double balance = sc.nextDouble();
        System.out.println();

        Account account;
        if (accountType.equals("Savings"))
        {
            account = new SavingsAccount(customerName,
                    accountNumber, accountType, balance);
        }
        else
        {
            account = new CurrentAccount(customerName,
                accountNumber, accountType, balance);
        }

        while(true)
        {
            System.out.println("1. Deposit 2. Display balance 3. Compute interest
                4. Withdraw 5. Exit");
            System.out.print("Enter choice");
            int choice = sc.nextInt();
            switch(choice)
            {
                case 1:
                    System.out.print("Enter amount to deposit : ");
                    double amount = sc.nextDouble();
                    account.deposit(amount);
                    break;
                case 2:
                    account.displayBalance();
                    break;
                case 3:
                    account.computeInterest();
                    break;
                case 4:
                    System.out.print("Enter amount to withdraw : ");
                    amount = sc.nextDouble();
                    account.withdraw(amount);
                    break;
                case 5:
                    sc.close();
                    System.exit(0)
                    break;
                default:
```

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

Output:

```
Enter customer name : Mithun G
Enter account number : 5528
Enter account type : Savings
Enter initial balance : 10000

1. Deposit
2. Display balance
3. Compute interest
4. Withdraw
5. Exit

* Enter Choice : 2
  Balance : 10000.0

* Enter Choice : 1
  Enter amount : 1000
  Amount deposited : 1000
```

```
* Enter choice : 4
  Enter amount to withdraw : 5000
  Amount withdrawn : 5000
```

Program:

Student.java

```
package CIE;
public class Student {
    public String USN;
    public String name;
    public int sem;
}
```

Internals.java

```
package CIE;
public class Internals extends Student {
    public int[] internalMarks = new int[5];
}
```

External.java

```
package SEE;
import CIE.Student;
public class External extends Student {
    public int[] externalMarks = new int[5];
}
```

GMain.java

```
import java.util.Scanner;
import CIE.Internals;
import SEE.External;

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

        Internals internalStudent = new Internals();
        System.out.println("Enter details");
        S.O.P("USN: ");
        internalStudent.USN = sc.nextLine();
        S.O.P("Name: ");
        internalStudent.name = sc.nextLine();
        S.O.P("Semester");
        internalStudent.sem = sc.nextInt();
        S.O.P("Enter marks");
        internalStudent.internalMarks = new int[5]
        for (int i = 0; i < 5; i++) {
            System.out.print("Course" + (i+1) + ": ");
            internalStudent.internalMarks[i]
        }
    }
    // Similar with External
    S.O.P("Internal marks:");
    printStudentDetails(internalStudent);
    S.O.P("External marks");
    printStudentDetails(externalStudent);

    Scanner.close();
}
```

```
private static void printStudentDetails(External e)
{
    S.O.p("USN:" + student.usn);
    S.O.p("Name:" + student.name);
    S.O.p("Sem:" + student.sem);
    S.O.p("Final marks:");
    for(int i=0; i<5; i++)
    {
        int finalMarks = student.internalMarks[i] + e.externalMarks[i];
        S.O.p("Course" + (i+1) + ":" + finalMarks);
    }
}
// Same with External Student
}
```

Output:

```
Enter details for Internal Student:
USN: 1BM22CS086
Name: Mithun
Sem: 3
Internal Marks:
Course 1: 75
Course 2: 80
Course 3: 85
Course 4: 90
Course 5: 75

Enter External Marks:
Course 1: 85
Course 2: 90
Course 3: 80
Course 4: 95
Course 5: 75
```

80

29/11/2024

Exception Handling

Program:

```
import java.util.Scanner

class WrongAge extends Exception {
    public WrongAge() {
        super("Invalid age:");
    }
}

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 {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAge();
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}

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

        try {
            S.o.p("Enter father's age");
            int fatherAge = sc.nextInt();
            S.o.p("Son's age");
            int sonAge = sc.nextInt();
            Father father = new Father(fatherAge);
            S.o.p("Father's age: " + father.getAge());
            Son son = new Son(fatherAge, sonAge);
            S.o.p("Son's age: " + son.getSonAge());
        } catch (WrongAge e) {
            S.o.p(e.getMessage());
        } catch (Exception e) {
            S.o.p("Invalid input");
        } finally {
            sc.close();
```

Output:
Enter father's age: 45
Enter son's age: 50
Invalid age!

Multithreading:
Program:

```
class DisplayThread extends Thread {
    private String message;
    private int interval;

    public DisplayThread(String message, int interval)
    {
        this.message = message;
        this.interval = interval;
    }
    public void run() {
        try{
            while(true) {
                S.o.p(message);
                Thread.sleep(interval *1000);
            }
        } catch(InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

```
public class ThreadDemo {
    p s v m(String[] args) {
        DisplayThread thread1 = new DisplayThread
                                ("BMSCE", 10);
        thread1.start();
        DisplayThread thread2 = new DisplayThread("CSE", 2);
        thread2.start();
    }
}
```

Output:
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering

Prof
19/2/24

Create label, button and TextField in a frame using AWT

```
import java.awt.*;
import java.awt.event.*;

public class AWTExample extends WindowAdapter
{
  Frame f;
  AWTExample()
  {
    f= new Frame();
    f.addWindowListener(this);
    Label l= new Label("Employee id:");
    Button b= new Button("Submit");
    TextField t= new TextField();
    l.setBounds(20,80,80,30);
    t.setBounds(20,100,80,30);
    b.setBounds(100,100,80,30);
    f.add(b);
    f.add(l);
    f.add(t);
    f.setSize(400,300);
    f.setTitle("Employee info");
    f.setLayout(null);
    f.setVisible(true);
  }
  public void windowClosing(WindowEvent e)
  {
    System.exit(0);
  }
  public static void main(String[] args)
  {
    AWTExample awt_obj= new AWTExample();
  }
}
```

2. Create a button and add a action listener for mouse click

```
import java.awt.*;
import java.awt.event.*;
public class EventHandling extends WindowAdapter implements
                ActionListener
  {
    Frame f;
    TextField tf;
    EventHandling()
    {
      f= new Frame();
      f.addWindowListener(this);
      tf= new TextField();
      tf.setBounds(60,50,170,20);
      Button b= new Button("Click me");
      b.setBounds(100,120,80,30);
      b.addActionListener(this);
      f.add(b);
      f.add(tf);
      f.setSize(300,300);
      f.setLayout(null);
```

```
}
public void actionPerformed(ActionEvent e)
{
  tf.setText("welcome");
}
public void windowClosing(WindowEvent e)
{
   System.exit(0);
}
public static void main(String args[])
{
  new EventHandling();
}
}
```