Quadratic Equation, LAB-1

```
import java.util.Scanner;

class Quadratic {
    int a,b,c;
    double r1,r2,d;

void input() {

Scanner s = new Scanner (System.in);
System.out.println(" Enter the coefficients of a,b,c ");
    a = s.nextInt();
    b = s.nextInt();
    c = s.nextInt();

}

void calc()

{ d = b*b - 4*a*c;

    if (d==0)
    { r1 = -b/(2.0*a);
      System.out.println(" Roots are real and equal");
      System.out.println(" Root1 = Root2 = " +r1);

    }
    else if (d>0)
    { ~~r1 = -b/(2.0*a);~~
      ~~System.out.println(" Roots are real & equal ")~~ distinct
      r1 = (-b + Math.sqrt(d)) / (2.0*a);
      r2 = (-b - Math.sqrt(d)) / (2.0*a);
      System.out.println(" Roots are real and distinct");
      System.out.println(" Root1 = " +r1 + ", Root2 = " +r2);

    }
```

```
else if (d<0)
  {  System.out.println("Roots are imaginary");
     r1 = -b/(2.0*a);
     r2 = Math.sqrt(Math.abs(d))/(2.0*a);
     System.out.println("Roots = " +r1+ "+i " +r2);
     System.out.println("Roots = " + r1 + "-i " +r2);
  }
}
}

Class QuadraticMain {
  public static void main(String[] args)
  {  Quadratic q = new Quadratic();
     q.input();
     q.calc();
  }
}
```

Output:

1) Enter the coefficients of a,b,c
1 5 6

Roots are real and distinct

Root1 = -2.0 , Root2 = -3.0

2) Enter the coefficients of a,b,c
10 2 3

Roots are imaginary

Roots = -0.1 + 10.5385

Roots = -0.1 - 10.5385

3) Enter the coefficients of a,b,c
2 4 2

Roots are real and equal

Root1 = Root2 = -1.0

Lab 9

```
import java.util.Scanner;

class student
{
    int usn;
    char name;
    int credits[];
    int marks[];
    int subjectcount;

Scanner s = new Scanner(System.in);
    void accept()
{
    System.out.println("Enter USN");
    int USN = s.nextInt();
    System.out.println("Enter name");
    char name = s.next();

    int credits= new int[subjectcount];
    int marks = new int[subjectcount];

        for (int i=0; i<subjectcount; i++)
        {
            SOP("Enter credit of each subject");
                Credit[i]= S.nextInt();
            SOP (" Enter marks of ever subject");
                marks[i]= S1.nextInt();
        }

    }

Void display()
    {   S.O.P(" USN=" + USN);
```

```
S.O.P ("Name =" + name);
for (int i=0; i< subject cnt; i++)
{   S.O.P (" Marks are = " + marks[i]);
    }
}

double gradecalc (int marks)
{   if (marks >=90)
    { return 10;
    }
    if (marks >=80 && marks <90)
    { return 9;
    }
    if (marks >= 70 && marks <80{
        return 8;
        }
    if (marks >=60 && marks <70)
    { return 7;
    }
    if (marks >= 50 && marks<60)
    { return 6;
    }
    if (marks <50)
    { return 0;
    }
    return 0;
}
```

```
float cale()
{ int totalgp=0;
  int totalc=0;
  float sgpa;
for (int i=0 ; i<= subcount ; i++)
{   totalgp+= gradeCale (marks [i] * credits [i])
    totalc += credits [i];

}
return (float) totalgp / totalc;
}
}

class Main
{
  public static void main (String[] args)
  {
      Student,S1 = new student();
        S2 .accept();
        S2. display ();
        S.O.P("SGPA of Student: " + S2. cale());
  }
}
```

Output:

Enter the USN of student:
087

Enter the name of the student:
Dheem

Enter no of subjects
4

Enter the credits of each subject:
4

Enter the marks of each subject:
92

Enter the credits of each subject:
3

Enter the marks of each subject:
87

Enter the credits of each subject:
3

Enter the marks of each subject:
78

Enter the credits of each subject:
2

Enter the marks of each subject:
95

USN: 087

Name: Dheeru

Marks are: 92

Marks are: 89

Marks are: 78

Marks are: 95

The SGPA of student is 9.25

08/01/24

Book database

```
import java.util.Scanner;

class Books

{
  String name, author;
  int price, num_page;

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

Scanner input = new Scanner(System.in)

Books()
{
  }

void accept()
{
  System.out.println("Enter book name: ");
  name = input.nextLine();
  System.out.println("Enter author name : ");
  author = input.next();
  System.out.print("Enter price:")
  price = input.nextInt;
  System.out.out.print("Enter no. of pages :");
  num_pages = input.nextInt();
  System.out.print
}
```

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

  return name + author + price + num_pages ;

  }

}

class BookRun
{
 public static void main (String [] args)
 {
  Scanner input = new Scanner (System.in) ;
  System.out.print (" Enter number of books : ");
    int n = input.nextInt();
  Books [] book = new Books [n];

  for (int i=0; i<n; i++)
  {
    System.out.println (" Book " + (i+1) + " : ");
    book [i] = new Books ();
    book [i].accept();

  }
  for (int i=0; i<n; i++)
  { System.out.println (" Book " + (i+1) + " : \n" +
                                book [i] + " \n");
  } input.close(); }
```

Output:

Enter number of books : 2

Book 1:

Enter name of book : abc

Enter name of author : dhee

Enter price of the book : 100

Enter no. of pages in the book : 100

Book 2:

Enter name of book : def

Enter name of author : dhee 2

Enter price of the book : 300

Enter no. of pages in book : 120.

Book 1:

Book name : abc

Author name : dhee

Price : 100Rs

Number of pages : 100 pages

Book 2:

Book name : def

Author name : dhee 2

Price : 300 Rs

Number of pages : 120 pages

Abstract class Shape.

```
import java.util.Scanner;

abstract class Shape
{
  double x, y;
  Shape (double x, double y)
  {
    this.x = x;
    this.y = y;
  }
abstract void printArea();
}

class Rectangle extends Shape
{
  Rectangle (double x, double y)
  {
    super (x, y);
  }
  void printArea()
  {
    System.out.println("Area of rectangle : " + (x*y) + "Sq units");
  }
}

class Triangle extends Shape
{
  Triangle (double x, double y)
  {
    super (x,y);
  }
```

```
void printArea()
{ System.out.println(" Area of Triangle :" + (0.5*x*y) + "sq.units");
  }
}
class circle extends Shape
{ circle (double x)
  { super(x, 0);
  }
  void printArea()
  { System.out.println("Area of Circle :" + (3.14*x*x) +
                                          "sq units ");
  }
}

class ShapeRun
{ public static void main (String[] args)
  { double x, y;
    Scanner input = new Scanner (System.in);
    System.out.println(" Enter length & width of rectangle : ");
    x = input.nextDouble();
    y = input.nextDouble();
    Rectangle r = new Rectangle (x, y);
      r.printArea();
    System.out.println(" Enter height and base of Triangle : ");
      x = input.nextDouble();
      y = input.nextDouble();
    Triangle t = new Triangle (x, y);
      t.printArea();
```

```
System.out.println(" Enter radius of circle: ");
    x = input.nextDouble();
    Circle c = new Circle(x);
    c.printArea();

  input.close();
 }
}
```

Output:

Enter length and width of Rectangle:
10
20

Area of Rectangle : 200.0 Squnits

Enter height and base of Triangle:
2
4

Area of Triangle : 4.0 Square units.

Enter radius of circle:
3.

Area of circle = 28.2599 sq.units.

22/01/24

Bank

LAB 5

```
import java.util.scanner;

class Account
{
  String customerName;
  long accountNumber;
  string accountType;
  double balance;

Account (StringcustomerName, long accountNumber, StringaccountType, double balance)
{ this.customerName= customerName;
  this.accountNumber = accountNumber;
  this.accountType  = accountType;
  this.balance = balance;

}

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

void displayBalance()
{ S.O.P("Acc.No: " + accNumber + " Name " + customerName + " Type " + accountType +
      " Balance " + balance);
}
}

class SavingsAccount extends Account
{
  SavingsAccount (String customerName, long accountNumber, double Balance)
  {
    super(customerName, accountNumber, "savings", balance);
  }
  void interest(double rate)
  { double interest = balance * rate/100;
    balance = balance + interest;
    S.O.P (" Interest computed & deposited. Updated balance : " + balance);
```

```
void withdraw(double amount)
{
  if (amount <= balance)
  {
    balance = balance - amount;
    S.O.P.("withdrawal of "+amount+" successful. Updated balance : " + balance);
  }
  else
  {
    System.out.println(" Insufficient balance");
  }
}
}
class CurrentAccount extends Account
{
  double minimumBalance;
  double serviceCharge;

  CurrentAccount(String customerName, long accountNumber, double balance, double minimumBalance,
                 double serviceCharge)
  {
    super(customerName, accountNumber, "Current", balance);
    this.minimumBalance = minimumBalance;
    this.serviceCharge = serviceCharge;
  }

  void checkMinimumBalance()
  {
    if (balance < minimumBalance)
    {
      balance = balance - serviceCharge;
      S.O.P.(" Minimum balance not maintained. Service charge imposed.
              Updated balance : " + balance);
    }
    else
    {
      S.O.P.(" Minimum balance maintained. No service charge.
              Updated balance " + balance);
    }
}
```

```
void cheque (double amount)
{
  balance = balance-amount;
  S.O.P ("withdrawal of " + amount + " successful. Updated balance: " + balance);
  }

}

public class Bank
{ public static void main (String[] args)
  {  Scanner S1 = new Scanner (System.in);

     S.O.P (" Savings Account:");
     S.O.P (" Enter customer name ");
       String name = S1.nextLine();

     S.O.P (" Enter acc no : ");

     long no = S1.nextLong();

     S.O.P (" Enter initial balance : ");

     double balance = S1.nextDouble();

     SavingsAccount SA = new SavingsAccount(name, no, balance);


     S.O.P (" Current Account ");
     S.O.P (" Enter customer name : ");
      name = S1.next();
     S.O.P (" Enter account number : ");
      no = S1.nextLong();
     S.O.P (" Enter balance:");
      balance = S1.nextDouble();
      S.O.P (" Enter min balance : ");
       double minBalance = S1.nextDouble();
       S.O.P (" Enter service charge : ");
        double charge = S1.nextDouble();
```

```
CurrentAccount CA = new CurrentAccount (name, no, balance, minBalance,
                                        charge);

S.O.P (" Enter deposit amount for SavingsAccount: ");
double SDA = SI.nextDouble();
SA.deposit (SDA);

S.O.P (" Enter interest rate for Savings Account : ");
double SIR = SI.nextDouble();
SA.interest (SIR);

S.O.P (" Enter withdrawal amount for Savings Account : ");
double SWA = SI.nextDouble();
SA.withdraw (SWA);

S.O.P (" Enter deposit amount for current Account : ");
double CDA = SI.nextDouble();
CA.deposit (CDA);

S.O.P (" Enter withdrawal amount for current Account : ");
double CWA = SI.nextDouble();
CA.cheque (CWA);

S.O.P (" Final Balances : ");
S.O.P (" Savings Account ");
SA.displayBalance();

S.O.P (" Current account : ");
CA.displayBalance ();
}

}
```

Output:

Savings Account:

Enter customer name: Dheen

Enter account number: 12345

Enter initial balance: 1000

Enter deposit amount for Savings Account: 500

Enter interest rate for Savings Account: 5

Enter withdrawal amount for Savings Account: 200

Current Account:

Enter customer name: Vamp

Enter account number: 54321

Enter balance: 2000

Enter minimum balance: 1000

Enter service charge: 20

Enter deposit amount for current Account: 800

Enter withdrawal amount for Current Account: 1200

Final Balances:

Savings Acount:

Account Number: 12345

Customer Name: Dheen

Account Type: Savings

Balance: 1305.0

Current Account:

Account Number: 54321

Customer Name: Vamp

Account Type: Current

Balance: 820.

LAB-6

Packages

CIE package: In CIE folder.

student.java:

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

Internals.java:

```
Package CIE;
import CIE.Student;
public class Internals extends Student
{
  public int[] InternalMarks;
  public Internals (String name, String usn, int sem, int[] InternalMarks)
  {
    super(name, usn, sem);
    this.InternalMarks = InternalMarks;
  }
```

SEE package : In SEE folder:

Externals.java :

```
package SEE;
import CIE.Student;

public class External extends Student
{ public int[] SeeMarks;
  public Externals (String name, String usn, int sem, int[] SeeMarks)
  { super(name,usn,sem);
    this.SeeMarks = SeeMarks;
  }
}
```

FinalMarks.java

```
import CIE.student;
import CIE.Internals;
import SEE.Externals;
import java.util.Scanner;

public class FinalMarks {
  public static void main (String[] args)
  { Scanner S1 = new Scanner(System.in);
    System.out.println(" Enter the number of students ");
    int n = S1.nextInt();

    String[] name = new String[n];
    String[] usn = new String[n];
    int[] Sem = new int[n];
    int[][] InternalMarks = new int[n][5];
    int[][] SeeMarks = new int[n][5];
```

```
for(int i=0; i<n; i++)
{ System.out.println(" Enter the details of Student " +
                    (i+1)+ " :");
  System.out.println("\n Name :");
  name[i] = sl.next();
  System.out.println("USN :");
  usn[i] = sl.next();
  System.out.println(" Sem: ");
  sem[i] = sl.nextInt();

  System.out.println(" Enter internal marks for 5 courses :");
  for(int j=0; j<5; j++)
  {  s.o.p(" Course " + (j+1)+ " :");
     InternalMarks[i][j] = sl.nextInt();
  }

  s.o.p(" Enter SEE marks for 5 courses :");
  for(int j=0; j<5; j++)
  { System.out.println("Course"+(j+1)+ " :");
    SeeMarks[i][j] = sl.nextInt();
  }.
}

int[][] FinalMarks = new int[n][5];
for(int i=0; i<n; i++)
{ Internals I1 = new Internals(name[i], usn[i],
                usn[i], sem[i], InternalMarks[i]);

  Externals E1 = new Externals(name[i], usn[i], sem[i],
                SeeMarks[i]);
```

```
int sum=0;
for(int j=0; j<5; j++)
{ FinalMarks[i][j] = I1.InternalMarks[i] + E1.SeeMarks[j];

}
System.out.println("\n Final Marks for " + n + " Student in
                    5 courses :");

for(int k=0; k<n; k++)
{   System.out.println(name[k] + ":");
    for(int j=0; j<5; j++) {
        sum = sum + FinalMarks[i][j];
    }
    System.out.println(sum);
}
}
s1.close();
}
}
```

Output:

Enter the number of students : 1

Enter details of student 1:

Name : John

USN : 087

Sem : 3

Enter internal marks for 5 courses:

course 1: 95

course 2: 95

course 3: 95

course 4: 95

course 5: 95

Enter See marks for 5 courses:

course 1: 90

course 2: 90

course 3: 90

course 4: 90

course 5: 90

Final Marks for 1 student in 5 courses:

John: 925

29/1/24

LAB - 7

WrongAge Exception handling

```
import java.util.Scanner
class WrongAge extends Exception
{ public WrongAge()
  { system.out.println(" Invalid age!");
  }

}
class Father
{ Father (int fage) throws WrongAge
  { if(fage<0)
    { throw new WrongAge(" Age cant be negative");
    }
  else
    { S.o.p(" Father age: "+fage);
    }
  }

}
class Son extends Father
{ Son (int fage, int sage) throws WrongAge
  { super(fage);
    if (sage >= fage)
    { throw new WrongAge(" Son age cant be greater than Father");
    }
  else
    { S.o.p (" Son age : +sage);
    }
  }
}

class Main{
    psvm(String[] a)
    Scanner s = new Scanner (System.in);
```

```
try {
    S.O.P ( " Enter fathers age : ");
  int fage = s.nextInt();

  S.O.P (" Enter Sons age : ");
  int sage = s.nextInt();
  Son son = new Son( fage, sage);
}
catch (WrongAge e)
  {  S.O.P (" Exception caught :" + e);

    }
  }
}
```

Output:

Enter fathers age : 40

Enter Sons age :

42

Son age can't be greater than Father

Exception caught: WrongAge

Enter fathers age : -10

Age can't be negative.

Exception caught: WrongAge.

Lab-8

Multiple Threads

```
class BMSCE extends Threads
{ public void run()
  {  while (true)
     { system.out.println ("BMS Engineering");
       try
       {  Thread.sleep (10000);
       }
       catch (InterruptedException e)
       {
          e.printStackTrace();
       }
     }
  }
}

class CSE extends Threads
{ public void run()
  { while (true)
    { S.O.P ("CSE");
      try
      {  Thread.sleep (2000);
      }
      catch (InterruptedException e)
      {
        e.printStackTrace();
      }
    }
  }
}
```

```
class Main
{
  public static void main (String [] args)
  {
    BMSCE b = new BMSCE();
      b.start();

    CSE c = new CSE();
      c.start();

  }

}
```

Output:

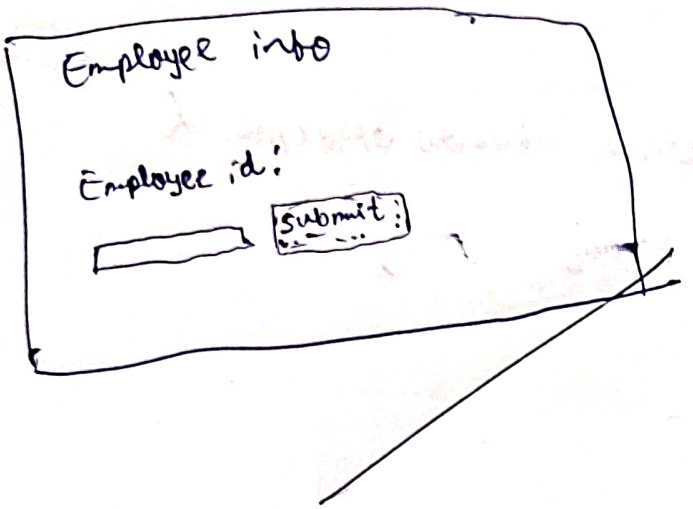
```
BMS Engineering
CSE
CSE
CSE
CSE
CSE
BMS Engineering.
CSE
CSE
CSE
CSE
CSE
BMS Engineering.
```

19.02.24

1. Creating 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();
  }
```

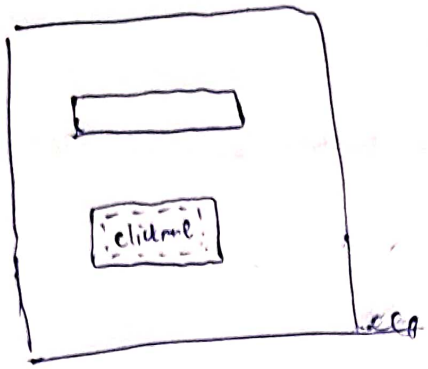
Output:

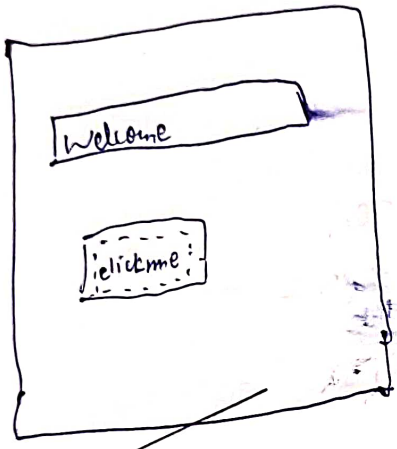


2. Create a button & add a action listener for Mouse click.

```
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);
f.setVisible(true);

}
public void actionPerformed(ActionEvent e)
{ tf.setText("Welcome");

}
public void windowClosing(WindowEvent e) {
System.exit(0);

}
public static void main(String args[]){
new EventHandling();
}
}
```

Output:





26.02.24