

Title

Page No.

Teacher's
Sign /
RemarksLAB - 1

Date _____

Page _____

Q Write a JAVA program that prints all Real solutions to the quadratic equation $ax^2+bx+c=0$. Read a, b, c and the use the quadratic formula. If the discriminant b^2-4ac is negative, display message that there is no real soln.

```

import java.util.Scanner;
public class quadratic {
    public static void main(String[] args) {
        float a, b, c, R1, R2;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter coeff. of x^2 : ");
        a = sc.nextFloat();
        System.out.println("Enter coeff. of x : ");
        b = sc.nextFloat();
        System.out.println("Enter constant : ");
        c = sc.nextFloat();
        if (a == 0) {
            System.out.println("Invalid input");
        } else {
            float det = (b * b) - 4 * a * c;
            if (det > 0) {
                R1 = (float) (-b + Math.sqrt(det)) / (2 * a);
                R2 = (float) (-b - Math.sqrt(det)) / (2 * a);
                System.out.println("R1 = " + R1);
                System.out.println("R2 = " + R2);
            } else if (det == 0)
                System.out.println("Roots are real and equal");
        }
    }
}
  
```

do float r = (float) ((-b) / (2 * a));

System.out.println ("R = " + r);

}

else

{ System.out.println ("The roots are imaginary
and distinct.");

float x = (float) ((-b) / (2 * a));

float y = (float) (Math.sqrt (-D) / (2 * a));

System.out.println ("R1 = " + x + " + " + y + " i");

System.out.println ("R2 = " + x + " - " + y + " i");

}

Output

enter coeff. of x^2 : 4

enter coeff of x : 6

enter constant : 2

the roots are : -0.5 and -1.0

enter coeff of x^2 : 1

enter coeff of x : 1

enter constant : 1

R1, R2 are imaginary and complex

$R_1 = 0.5 + 1.5i$

$R_2 = 0.5 - 1.5i$

$\begin{cases} R_1 \\ R_2 \end{cases}$

LAB

Q

Develop a JAVA
student, with
credit and me-
display detail

↳

import java.util.*;

public class student

String name

String name

public int [] credit

public int [] marks

public void

Scanner

System

wn =

System

name

System

for (

sy

;

c

s

ynb

LAB-2

Open
Page

Q Develop a JAVA program to create a class Student, with members usn, name and an array credit and marks. Include method to accept and display detail and a method to calculate CGPA.

↳ import java.util.Scanner;

public class Student {

String usn;

String name;

public int[] credits = new int[8];

public int[] marks = new int[8];

public void acceptdetails() {

Scanner s = new Scanner(System.in);

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

usn = s.nextLine();

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

name = s.nextLine();

System.out.println("Enter marks for each subject : ");

for (int i=0; i<credits.length; i++) {
System.out.println("Enter credit for subject " + (i+1) + ": ");

credit[i] = s.nextInt();

System.out.println("Enter marks for subject " + (i+1) + ": ");

marks[i] = s.nextInt();

public double calcCGPA() {

public int total_credits = 0;

public int total_grade = 0;

double ans;

int grade_point;

P.T.O

```

for (int i = 0; i < credits.length; i++) {
    total_credits += credits[i];
}
int gpa;
grade_points = (mark[i] / 10) + 1;
if (grade_points == 1)
    grade_points = 10;
else if (grade_points < 6)
    System.out.println("GPA cannot be calculated as there is back in a subject");
return;
}
totalGrade += grade_points;

```

```
ans = totalGrade / total_Credits;  
System.out.println("GPA is : "+ ans);  
  
public class GPA {  
    public static void main(String[] args) {  
        Scanner s = new Scanner(System.in);  
        Student student = new Student();  
        student.acceptDetails();  
        System.out.println("Student Details :");  
        System.out.println("USN : " + student.usn);  
        System.out.println("Name : " + name);  
        System.out.println  
        student.CalcGPA();  
    }  
}
```

Output

Enter USN : ABC

Enter name : K

Enter marks for ea

Enter credits for su

Enter marks for sl

Enter credits for ..

Enter marks for ..

Enter credits ..

Enter marks ..

Enter credits ..

Enter marks ..

Enter credits ..

U marks ..

U credits ..

student details

USN : ABC

Name : K

SUPA is : 9.

length ; i++)
details[i];

marks[i] / 10) + 1,
)

0. 4

<6) {

n ("SGPA cannot be calculated
there is back in a subject")

- points;

total_credits:

SGPA is : " + ans);

String[] args) {

(System.in);

student();

st_Details : " +

" + student.usn)

me : " + name);

Output

Enter USN : ABC

Enter name : K

Enter marks for each subject :

Enter credits for subject 1 : 4 By

Enter marks for subject 1 : 91

Enter credits for subject 2 : 3 Y

Enter marks for subject 2 : 93

Enter credits 11 3 : 3

Enter marks 11 3 : 89

Enter credits 11 4 : 3

11 marks 11 4 : 83

11 credits 11 5 : 3

11 marks 11 5 : 85

11 credits 11 6 : 21

11 marks 11 6 : 95

11 credits 11 7 : 21

11 marks 11 7 : 93

11 credits 11 8 : 1

11 marks 11 8 : 92

student details :

USN : ABC

Name : K

SGPA is : 9.55

LAB-B

Q

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

```
import java.util.Scanner;
class Book {
    String name;
    String author;
    int price;
    float numPages;
    void setDetails() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter book name, author, price, num-pages : ");
        name = sc.nextLine();
        author = sc.nextLine();
        price = sc.nextInt();
        numPage = sc.nextFloat();
    }
}
```

```
void getDetails() {
    String details = toString();
    System.out.println(details);
}
```

```
public String toString() {
    return "The book " + name + " was written by "
    + author + " consisting of " + numPages + " pages and "
    "costs " + price + " ;
```

Y

```
public static void main() {
    Scanner S = new Scanner
    System.out.println("Enter no. of books");
    int n = S.nextInt();
    Book b = new Book[n];
    for (int i = 0; i < n; i++) {
        b[i] = new Book();
        b[i].setDetails();
    }
}
```

```
System.out.println("book details");
for (int i = 0; i < n; i++)
    b[i].getDetails();
}
```

Output :
 Enter no. of books : 1
 Enter book name, author, price, num-pages : XYZ ABC 1000 50
 The book XYZ was written by ABC consisting of 50 pages and costs 1000.

AB-3

Book which contains 4 members
price, num-pages. Include
set values for the members.
Add to set and get details of
use a toString() method that
the complete details of the book.
m to create n book objects.

Scanner:

pages;
sc() <
new Scanner(System.in);
println("Enter book name, author,
price, no. of pages : ");
nextLine();
nextLine();
nextInt();
sc.nextLine();

{
= toString();
ln(details);

ring() &
" + name + " was written by "
ring & " + num-pages + " pages and

Date _____
Page _____

```
public static void main (String[] args)
{
    Scanner s = new Scanner (System.in);
    System.out.println ("Enter no. of books : ");
    int n = s.nextInt ();
    Book b = new Book [n];
    for (int i = 0; i < n; i++) {
        b[i] = new Book ();
        b[i].setDetails ();
    }
    System.out.println ("book details");
    System.out.println ();
    for (int i = 0; i < n; i++) {
        b[i].getDetails ();
    }
}
```

Output :

Enter no. of books ? 1

Enter book name, author, price and no. of pages :

X42 ABG 1000 50

The book X42 was written by ABG costing 1
50 pages and costs 1000.

LAB 4

Q

Develop a JAVA program to develop an abstract class shape that contains two integers and an empty method printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the class extends the class Shape. Each one of the class contains only one method printArea().

```

import java.util.Scanner;
class shape {
    int a;
    int b;
    void printArea()
    abstract class Shape {
        int x, y;
        abstract void area();
        public static void main (String [] args)
        {
            Shape obj_1 = new Circle ();
            Shape obj_2 = new
                obj_.area ();
            Shape obj_2 = new Rectangle ();
            obj_2.area ();
            Shape obj_3 = new triangle ();
            obj_3.area ();
        }
    }
}

```

y

class rectangle extends shape {

void rectangle () {

Scanner sc = new Scanner (System.in);

System.out.println ("Enter length and breadth")

x = sc.nextInt ();

, y = sc.nextInt ();

void area () {

System.out.println ("Area = " +

y

class triangle extends shape {

void triangle () {

Scanner sc = new Scanner

System.out.println ("Enter")

X = sc.nextInt ();

Y = sc.nextInt ();

y

void area () {

System.out.println ("Area = " +

y

class circle extends shape {

void circle () {

Scanner sc = new Scanner

System.out.println ("Enter")

X = sc.nextInt (), Y =

y

void area () {

System.out.println ("

y

Output

Enter length and breadth

Area is 24

Enter base and altitude

Area is 2

Enter radius : 2

Area is 25.12.

AB 4

JAVA program to develop an abstract class that contains two integers and an `printArea()`. Provide three classes, Rectangle, Triangle and Circle such that the class extends the class Shape. The class contains only one method `area()`.

util. Scanner;

area()
shape{

void area();
c void main (String [] args)

= new Circle();

x = 0

);

2 = new Rectangle();

2-area();

3 = new Triangle();

);

and shape 1

Scanner (System.in);
ln ("Enter length and breadth")

;

void area () {

System.out.println ("Area of rectangle is " + x * y);
y

Class triangle extends shape {

void triangle () {

Scanner sc = new Scanner (System.in);

System.out.println ("Enter base & altitude ");

x = sc.nextInt();

y = sc.nextInt();

y

void area () {

System.out.println ("Area is " + x * y * 0.5);
y

Class circle extends shape {

void circle () {

Scanner sc = new Scanner (System.in);

System.out.println ("Area Enter radius ");

x = sc.nextInt(); z = y = x;

y

void area () {

System.out.println ("Area is " + 2 * 3.14 * x * y);
y

Output

Enter length and breadth : 4 6

Area is 24

Enter base and altitude : 2 2

Area is 2

Enter radius : 9

Area is 25.12.

LAB-5

Q Create a package CIE which has two classes - Student and Internals. Class Student has members like usn, name, sum. The class Internals has an array that stores internal marks scored in 5 subjects of current semester of the student. Create another package which has class External which is a derived class of Student. This class has an array after that stores the SEM marks scored in 5 semesters of the current semester. Import the two packages in a file that declares the final marks of n students.

L ↳ Student.java:

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

I ↳ Internals.java:

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

P.T.O

public Internals (string usn, string name, int sum) {
 super (usn, name, sum);
 this.m = m;
}

Y ↳ External.java:

```
package SEE;
import CIE.Student;
public class External extends Student {
    public int sm[] = new int[5];
    public External (string usn, string name, int sum) {
        super (usn, name, sum);
        this.sm = sm;
    }
}
```

Y ↳ mainclass.java:

```
import java.util.Scanner;
import CIE.Student;
import CIE.Internals;
import CIE.SEE.External;
public class mainclass {
    public static void main () {
        int tm = 0;
        Scanner in = new Scanner (System.in);
        System.out.println ("Enter usn, name, sum");
        usn = in.nextLine();
        name = in.nextLine();
        sum = in.nextInt();
        Student s = new Student (usn, name, sum);
        Internals i = new Internals ();
        External e = new External (usn, name, sum);
        System.out.println ("Enter 5 internal marks");
        for (int j = 0; j < 5; j++) {
            m[j] = in.nextInt();
        }
        System.out.println ("Enter 5 sem marks");
        for (int j = 0; j < 5; j++) {
            sm[j] = in.nextInt();
        }
        System.out.println ("Final result is " + s.sum + " " + i.sum + " " + e.sum);
    }
}
```

LAB-5

Java CSE which has two classes - Student & Class. Student has members like id, name, marks scored in 5 subjects of semester. Create another class External which is a class of Student. This class has an array stores the SEM marks stored in 5 subjects of semester. Import the two packages & declares the final marks of n.

```
Student {  
    id, name;  
    int sum;  
    int (string id, string name, int sum)  
  
    id = id;  
    name = name;  
    sum = sum;
```

```
class extends Student{  
    int sm[];  
    sm = new int[5];  
    f T O
```

Date _____
Page _____

```
public Internal (string id, string name, int sum,  
int []sm )
```

```
{  
    super (id, name, sum);  
    this.sm = sm;
```

External.java :

```
package SEE;  
import CSE.Student;  
public class External extends Student {  
    public int sm[] = new int [5];  
    public External (string id, string name, int sum,  
    int []sm )
```

```
{  
    super (id, name, sum);  
    this.sm = sm;
```

mainclass.java

```
import java.util.Scanner;  
import CSE.Student;  
import CSE.Internal;  
import SEE.External;  
public class mainclass {  
    public static void main (String args[]){  
        int tm = 0;  
        Scanner in = new Scanner (System.in);  
        System.out.println ("Enter no. of student: ");
```

```

int n = in.nextInt();
Internals [] im = new Internals [n];
External [] em = new External [n];
student [] stu = new student [n];
for (int i=0; i<n; i++) {
    System.out.println ("Enter detail for student"
                        + (i+1) + ":");
    System.out.print ("Enter name: ");
    in.nextLine();
    String name = in.nextLine();
    System.out.print ("Enter USN: ");
    String usn = in.nextLine();
    System.out.print ("Enter semester: ");
    int sem = in.nextInt();
    int [] internal_marks = new int [5];
    int [] external_marks = new int [5];
    System.out.println ();
    System.out.println ("Enter Marks details:");
    for (int j=0; j<5; j++) {
        System.out.print ("Enter internal marks"
                            " for course " + (j+1) + ":" );
        internal_marks [j] = in.nextInt();
        System.out.print ("Enter external marks"
                            " for course " + (j+1) + ":" );
        external_marks [j] = in.nextInt();
    }
    System.out.println ();
    stu[i] = new student (usn, name, sem,
                          internal_marks);
    im[i] = new Internals (usn, name, sem,
                          internal_marks);
    em[i] = new External (usn, name, sem,
                          external_marks);
}

```

```

in.nextInt();
[] im = new Internal[5];
[] em = new External[5];
[] stu = new Student[5];
for (int i=0; i<n; i++) {
    out.println("Enter detail for student" + (i+1) + ":");
    out.print("Enter name: ");
    name = in.nextLine();
    out.print("Enter usn: ");
    usn = in.nextLine();
    out.print("Enter semester: ");
    m = in.nextInt();
    internal_marks = new int[5];
    external_marks = new int[5];
    out.println();
    out.println("Enter Marks details:");
    for (int j=0; j<5; j++) {
        out.println("Enter internal marks for course " + (j+1) + ":");
        imarks[j] = in.nextInt();
        out.println("Enter external marks for course " + (j+1) + ":");
        emarks[j] = in.nextInt();
    }
    out.println();
    new Student(usn, name, sem, internal_marks);
    new Internal(usn, name, sem, internal_marks);
    new External(usn, name, sem, external_marks);
}

```

P.P.O

```

* System.out.println("Final marks Details:");
for (int i=0; i<n; i++) {
    System.out.println("Student " + (i+1) + ":"),  

    System.out.println("Name: " + stu[i].name),
    System.out.println("USN: " + stu[i].usn),
    System.out.println("Sem: " + stu[i].sem),
    for (int j=0; j<5; j++) {
        tm += 10 * im[i].m[j] + em[i].em[j];
    }
    System.out.println("Final marks of course" + (i+1) + ":" + tm);
}

tm = 0;
System.out.println();
System.out.println();
System.out.println("Output →");
Enter no. of student: 2
Enter details for student 1
enter Name: Kshitiz
enter USN: 1234
enter Semester: 2
Enter marks details:
Enter Internal marks for courses: 34
Enter External marks 11 11 : 98
Enter Internal 11 11 2 : 89
Enter External 4 1 11 2 : 97
Enter Internal 11 4 11 3 : 83
Enter External 4 11 3 3 : 86
Enter Internal 11 4 11 4 : 8433
Enter External 11 4 11 4 : 64
Enter Internal 4 1 11 5 : 40
Enter External 4 1 11 5 : 90

```

Enter details for student 2:

Final marks details:

Student 1

Name : kishore

USN : 1234

Semester : 2

Final marks for course 1 : 132

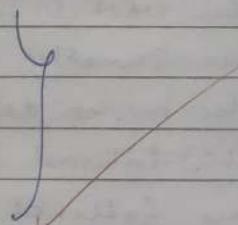
4 2 : 136

4 3 : 89

6 4 : 93

6 5 : 130

Student 2 :



LAB-6

Q Write a program that
exception. Create a base
implement a constructor
throws the exception.
Input age < 0. In a class
that uses base function
throws exception if

↳ import java.util.*;
class WrongAge extends
WrongAge Exception
System.out.println("Age
is less than zero")

↳ Use Father &
In age.
Father(int
age)

↳ This age
if age <
throw new
Exception("Age
is less than zero")

↳ else &
System.out.println("Age
is greater than zero")

↳ class Son extends
int sonAge
Son(int age)
System.out.println("Age
is greater than zero")

details for student 2:

marks details:

marks for course 9 : 132
2 : 136
3 : 89
4 : 93
5 : 130

LAB-6

Date _____
Page _____

Q Write a program that demonstrates handling of exception. Create a base class. In Father class, implement a constructor that takes the age and throws, throws the exception WrongAge() where Input age < 0. In Son class, implement a constructor that uses both father's and son's age and throws exception if son's age is \geq father's age.

```
import java.util.*;  
class WrongAge extends Exception {  
    WrongAge (String msg) {  
        System.out.println(msg); super(msg);  
    }  
}  
class Father {  
    int age;  
    Father (int age) throws WrongAge {  
        this.age = age;  
        if (age < 0) {  
            throw new WrongAge ("Age can't  
            be less than 0");  
        }  
    }  
    else {  
        System.out.println ("Father's age verified");  
    }  
}  
class Son extends Father {  
    int sonage;  
    Son (int age, int sonage) throws WrongAge {  
        Super (age);  
        this.sonage = sonage;  
    }  
}
```

```
if (sonage < 0 || sonage >= age)
```

```
{
```

```
    throw new WrongAgeException ("Son's age  
    invalid");
```

```
else {
```

```
    System.out.println ("Son's age verified");
```

```
}
```

```
}
```

```
class Age {
```

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

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

```
        int age, sonage;
```

```
        System.out.println ("Enter father's age : ");
```

```
        age = in.nextInt();
```

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

```
        sonage = in.nextInt();
```

```
    try {
```

```
        Son son1 = new Son (age, sonage);
```

```
}
```

```
    catch (WrongAgeException e1) {
```

```
        System.out.println (e1);
```

```
}
```

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

```
        System.out.println (e);
```

```
}
```

```
}
```

↳ Output

Enter father's age : 42

Enter son's age : 24

Father's age verified!

Son's age verified

Enter father's age : -10

Enter son's age : 12

Age can't be less than 0

Wrong Age Exception

Enter father's age : 24

Enter son's age : 42

Father's age verified.

Son's age is invalid

Wrong Age Exception.

8/29/11

$<= 11 \text{ sonage} \geq \text{age})$

new WrongAgeException ("Son's age is invalid");

out.println ("Son's age verified");

```

    void main (String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print ("Enter father's age : ");
        int fage = sc.nextInt();
        System.out.print ("Enter son's age : ");
        int sage = sc.nextInt();
        if (sage <= 11 || sage >= fage) {
            throw new WrongAgeException ("Son's age is invalid");
        }
    }
}

```

```

    catch (WrongAgeException e) {
        System.out.println (e.getMessage());
    }
}

```

→ Output

Enter father's age : 42

Enter son's Age : 24

Father's age verified!

Son's Age verified

Enter father's age : -10

Enter son's Age : 12

Age can't be less than 0

Wrong Age Exception

Enter father's Age : 24

Enter son's Age : 42

Father's Age Verified.

Son's Age is Invalid

Wrong Age Exception.

8/29/11

LAB - 7

Q Sample program in threads.

① Class Thread1

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

```
    Thread t = Thread.currentThread();  
    System.out.println ("CT: " + t);
```

```
    try
```

```
    { for (int n=5; n>0; n--)
```

```
        System.out.println (n);
```

```
        Thread.sleep(500);
```

```
}
```

```
    catch (InterruptedException ie)
```

```
        System.out.println ("The sleeping thread  
is woken up");
```

Output

```
↳ CT : Thread [main, 5, main]
```

```
CT: Thread (Current main Thread, 5, main)
```

```
5
```

```
4
```

```
3
```

```
2
```

```
1
```

② Class NewThread implements
of

```
Thread t;  
NewThread()
```

```
t = new Thread (this, "  
System.out.println ("CT  
t.start();
```

```
public void run()
```

```
try
```

```
for (int n=5; n>0;
```

```
System.out.println  
Thread.sleep(500);
```

```
}
```

```
catch (InterruptedException ie)
```

```
System.out.println
```

```
System.out.println ("
```

Class Thread2

```
public static void m
```

```
new NewThread
```

```
System.out.p
```

8-7

In threads.

```
void main (String ss[])
```

```
    Thread currentThread();  
    println ("CT: " + t);
```

```
n=5; n>0; n-- )
```

```
m.out.println(n);  
t.sleep(500);
```

```
interruptedException ie)
```

```
ut.println ("The sleeping thread  
is woken up");
```

```
n, 5, main]
```

```
ent main Thread, 5, main ]
```

② class NewThread implements Runnable

```
Thread t;
```

```
NewThread()
```

```
t = new Thread (this, "NThread");  
System.out.println ("CT" + t);  
t.start();
```

```
public void run()
```

```
try
```

```
for (int n=5; n>0; n--)
```

```
System.out.println ("child: " + n);  
Thread.sleep(500);
```

```
}
```

Catch (InterruptedException ie)

```
if
```

```
System.out.println ("Child Thread Interrupted");
```

```
}
```

```
System.out.println ("Child Thread quitting");
```

```
class Thread2
```

```
public static void main (String ss[])
```

```
{
```

```
New NewThread();
```

```
System.out.println ("Back in main");
```

```

try
{
    for (int n=5; n>0; n--)
    {
        System.out.println("Main Thread  
Interrupted");
    }
    System.out.println("Main Thread  
quitting");
}

Output
[CT Thread [NThread, 5, main]
Back in main
child : 5
Main Thread : 5
child : 4
Main Thread : 4
child : 3
child : 2
Main Thread : 3
child : 1
Child Thread quitting
Main Thread : 2
Main Thread : 2
Main Thread quitting

```

Q Write a program that creates a thread displaying "BMS Co" once every 10 seconds and "CSF" once every two seconds.

↳ class NewThread implements Runnable

Thread t;
NewThread()

t = new Thread(this);
System.out.println(t.start());

public void run()

try
{
 for (int n=10; n>0; n--)
 {
 System.out.println("BMS Co");
 Thread.sleep(10000);
 System.out.println("CSF");
 Thread.sleep(2000);
 }
}

Catch (InterruptedException)

System.out.println("Error");

System.out.println("Program Terminated");

Q Write a program that creates 2 threads, one thread displaying "BMS College of Engineering" once every 10 seconds and another displaying "CSE" once every two seconds.

↳ class NewThread implements Runnable

{

 Thread t;

 NewThread()

{

 t = new Thread(this, "NThread");

 System.out.println("CT" + t);

 t.start();

}

 public void run()

 try

 {

 for (int n=10; n>0; n--)

 {

 System.out.println("CSE");

 Thread.sleep(2000);

}

}

 catch (InterruptedException ie)

{

 System.out.println("Child Thread Interrupted");

}

 System.out.println("Child Thread quitting");

}

Class Threads

```
public static void main(String args)
{
    new NewThread(),
    System.out.println("Back in Main"),
    try
    {
        for(int n=2; n>0; n--)
        {
            System.out.println("B.M.S.C.E"),
            Thread.sleep(10000);
        }
    }
    catch(InterruptedException ie)
    {
        System.out.println("Main Thread quitting"),
        System.out.println("Main Thread quitting");
    }
}
```

Output

↳ CTThread [NThread, 5, main]

Back in main

B.M.S.C.E

CSE

CSE

CSE

CSE

B.M.S.C.E

CSE

CSE

CSE

CSE

~~REMOVING~~

Main Thread quitting
Child Thread quitting

CSE

REVIEW

Main Thread quitting

Child Thread quitting

Date _____
Page _____

LAB-8

Q Develop a JAVA program to create a class Bank that maintains 2 kinds of account for its customer, one called Savings account and the other Current account. The savings account provides compound interest and withdraw 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 classes Current and Savings to make them more specific to their requirements.

- a) Accept deposit from customer and update balance
- b) Display the balance
- c) Compute and deposit interest
- d) Permit withdraw and update balance

↳ Import java.util.Scanner;

Class Account {

```
String customerName;
int accountNumber;
String accountType;
double balance;
```

RTO

Account (String name, int accNo)

```
{  
    customerName = name;  
    accountNumber = accNo;  
    accountType = type;  
    balance = bal;
```

}

```
void deposit (double amount)  
{  
    balance += amount;  
    System.out.println ("Deposited");  
}
```

}

```
void displayBalance () {  
    System.out.println ("Account balance is " + balance);  
}
```

```
void withdraw (double amount)  
{  
    if (balance - amount >= minimumBalance)  
        balance -= amount;  
    else  
        System.out.println ("Insufficient balance");  
}
```

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

}

Y

```
class Current extends Account {  
    double minimumBalance;  
    double serviceCharge;  
    Current (String name,  
            double bal, double minBal,  
            double svcChg)  
    {  
        super (name, accNo, bal, minBal, svcChg);  
    }
```

```
super (name, accNo, bal, minBal, svcChg);
```

LAB-5

Date _____
Page _____

Java program to create a class
maintains 3 kinds of account
owner, one called savings account
other current account. The savings
provides compound interest and withdraw
w/o cheque book facility. The
savings provides cheque book facility
rest. Current account holders
maintain a minimum balance
balance falls below this level,
charge is imposed.

an account that stores customer
name, account number, and type of account.
use classes Current and Savings
more specific to their

deposit from customer and update

balance

deposit interest

and update balance

Scanner;

customerName;
accountNumber;
accountType;
balance;

RTO

Account (String name, int accNo, String type, double bal)

{
 CustomerName = name;
 accountNumber = accNo;
 accountType = type;
 balance = bal;

y
void deposit (double amount) {
 balance += amount;

 System.out.println ("Deposit of Rs. " + amount +
 ". " + "successful");

y
void displayBalance () {

 System.out.println ("Account Balance : Rs. " + balance);

y
void withdraw (double amount) {
 if (balance - amount >= 0) {

 balance -= amount;

 System.out.println ("Withdrawal of Rs. " +
 amount + " successful");

y
else {

 System.out.println ("Insufficient balance
 for withdrawal");

class Current extends Account {

 double minimumBalance;

 double serviceCharge;

 Current (String name, int accNo, String type,
 double bal, double minBal, double charge)

{
 super (name, accNo, type, bal);

minimumBalance = minBal

ServiceCharge = charge;

y void withdraw(double amount) {

if (balance - amount >= minimumBalance)

balance -= amount;

System.out.println ("Withdrawal of Rs. " +

amount + "Successful");

else {

System.out.println ("Insufficient balance for
withdrawal. Service charge of Rs. " +
ServiceCharge + "Applied.");

balance -= serviceCharge;

y

y void checkbook() {

System.out.println ("Checkbook facilities are
available and will be sent soon.");

y

class SavAcc extends Account {

double interestRate;

SavAcc (String name, int accNo, String type,
double bal, double rate)

super (name, accNo, type, bal);

interestRate = rate;

y void computeInterest() {

double interest = balance * (interestRate / 100);

balance += interest;

System.out.println ("Interest of Rs. " + interest
+ "Added to account")

y void checkbook()

System.out.println ("not av")

y public class bank {

public static void main (CA
currAcct amount) {

new currAcct ("A",
5000);

SavAcc SA = new SavAcc

System.out.println ("Cur
CA. displayBalance());

CA. deposit (2000);

CA. displayBalance();

CA. withdraw (7000);

CA. displayBalance();

CA. withdraw (3000);

CA. displayBalance();

CA. displayBalance();

SA. computeInterest();

SA. displayBalance();

SA. withdraw (15000);

SA. displayBalance();

CA. checkbook();

Balance = minBal
charge = charge;
withdraw(double amount) {
balance - amount >= minBalance}

area -= amount;
cout.out.print("Withdrawal of Rs. " +
amount + " Successful");

cout.out.println("Insufficient balance for
withdrawl. Service charge of Rs. " +
serviceCharge + " applied.");
charge = serviceCharge;

else {
cout.out.println("Chequebook facilities are
available soon");

extends Account {

interestRate;
name, int accNo, string type,
bal, double rate)

me, accNo, type, bal);

interest();

interest();

it = balance * (interestRate / 100);
interest \$
when ("Interest of Rs. " + interest
\$ to account ");

Date _____
Page _____

y
void Chequebook() {
if
System.out.println("Chequebook facilities
not available");
y
public class bank {
public static void main (String args[]) {
CA currAcct accountAccount =
new currAcct ("Name", 123456, "Current",
5000, 1000, 50);
SA savAcct = new SA ("Kshwir", 654321,
"Savings", 10000, 5);
System.out.println("Current Account Details: ");
CA.displayBalance();
CA.deposit(2000);
CA.displayBalance();
CA.withdraw(7000);
CA.displayBalance();
CA.withdraw(3000);
CA.displayBalance();
CA.chequebook();
System.out.println("In Savings A/c: ");
SA.displayBalance();
SA.deposit(5000);
SA.displayBalance();
SA.computeInterest();
SA.displayBalance();
SA.withdraw(15000);
SA.displayBalance();
CA.displayBalance();
CA.chequebook();

Output giving Account Details :

↳ Account Balance = Rs 15000.0

Deposit of Rs. 200.0 successful

Account Balance : Rs 15200.0

Interest Rate of Rs 700.0 added to account

Account Balance : Rs 15700.0

Withdraw of Rs 1500.0 successful

Account Balance : Rs 200.0

Checkbook facilities not available.

~~Q 19/10~~

① ButtonDemo.java

```
import java.awt.*
```

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

```
import java.applet.*
```

```
public class Button
```

```
String msg = "
```

```
Button yes, no
```

```
yes = new Bu
```

```
no = new Bu
```

```
maybe = new B
```

```
add(yes);
```

```
add(no);
```

```
add(maybe);
```

```
yes.addActionListener(new A)
```

```
no.addActionListener(new A)
```

```
maybe.addActionListener(new A)
```

```
public void
```

```
String str
```

```
y.addActionListener(new A)
```

```
msg
```

```
y
```

```
use if
```

```
msg
```

```
y
```

```
else if
```

```
msg
```

LAB-9

AWT

Date _____
Page _____

(1) ButtonDemo.java

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

```
public class ButtonDemo extends Applet implements
    ActionListener {
```

```
    String msg = " ";
    Button yes, no, maybe;
    yes = new Button("Yes");
    no = new Button("No");
    maybe = new Button("Undecided");
    add(yes);
    add(no);
    add(maybe);
    yes.addActionListener(this);
    no.addActionListener(this);
    maybe.addActionListener(this);
    public void actionPerformed(ActionEvent ae) {
```

```
        String str = ae.getActionCommand();
```

```
        if (str.equals("Yes")) {
```

```
            msg = "You pressed Yes";
```

```
}
```

```
        else if (str.equals("No")) {
```

```
            msg = "You pressed No";
```

```
}
```

```
        else {
```

```
            msg = "You pressed Undecided";
```

```
}
```

```
        repaint();
```

```
}
```

Account Details :

Balance : Rs 1000.0

Rs. 200.0 successful

Rs 1000.0

Rs 200.0 added to account

Balance : Rs 1500.0

Rs 1500.0 successful

Balance : Rs 200.0

Balance not available.

8/19/20

```
public void paint(Graphics g) {
```

 g.drawString("Hello", 6, 100);

 y

② ButtonList.java

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

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

```
public class ButtonList extends Frame implements
```

ActionListener {

```
    String msg = "Hello";
```

```
    Button butt[] = new Button[3];
```

```
    public ButtonList() {
```

```
        setLayout(new FlowLayout(1));
```

```
        Button yes = new Button("Yes");
```

```
        Button no = new Button("No");
```

```
        Button maybe = new Button("Undecided");
```

```
        add(componentComplete());
```

```
    butt[0] = (Button) add(yes);
```

```
    butt[1] = (Button) add(no);
```

```
    butt[2] = (Button) add(maybe);
```

```
    for (int i=0; i<3; i++)
```

```
        butt[i].addActionListener(this);
```

```
    addWindowListener(new MyWindowAdapter());
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        for (int i=0; i<3; i++) {
```

```
            if (e.getSource() == butt[i]) {
```

```
                msg = "You pressed " + butt[i].getLabel();
```

 y

```
    public void paint(Graphics g) {
```

```
        g.drawString(msg, 20,
```

 y

```
    public static void main
```

```
        ButtonList aa = new B
```

```
        aa.setTitle("ButtonList");
```

```
        aa.setVisible(true);
```

```
    }
```

```
    class MyWindowAdapter extends WindowAdapter {
```

```
        public void windowClosed
```

```
            System.exit(0);
```

③ DivisionMain.java

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

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

```
public class Division implements ActionListener
```

```
    {
```

```
        TextField num1, nu
```

```
        Button d, result;
```

```
        Label outResult;
```

```
        String out = " ";
```

```

    void paint (Graphics g) {
        repaint();
    }

    public void paint (Graphics g) {
        g.drawString ("msg", 20, 10);
    }

    public static void main (String args[]) {
        ButtonList aa = new ButtonList();
        aa.setSize (new Dimension(200, 150));
        aa.setTitle ("ButtonList");
        aa.setVisible (true);
    }

    class MyWindowAdapter extends WindowAdapter {
        public void windowClosing (WindowEvent we) {
            System.exit (0);
        }
    }

    class Main {
        Dimension Main.java
    }

    import java.awt.*;
    import java.awt.event.*;
    public class Division Main extends Frame
        implements ActionListener {
        JTextField num1, num2;
        Button d, result;
        Label outResult;
        String out = " ";
        }

        Division () {
            num1 = new JTextField("1");
            num2 = new JTextField("2");
            d = new JButton ("Division");
            result = new JButton ("Result");
            outResult = new JLabel ("");
            outResult.setFont (new Font ("Serif", 1, 16));
            outResult.setHorizontalAlignment (JLabel.CENTER);
            outResult.setForeground (Color.BLUE);
            outResult.setOpaque (true);
            outResult.setBackground (Color.LIGHT_GRAY);
            outResult.setBorder (BorderFactory.createEtchedBorder ());
            outResult.setPreferredSize (new Dimension (200, 50));
            outResult.setLayout (new GridLayout (1, 1));
            outResult.add (out);
            setLayout (new GridLayout (2, 1));
            add (num1, "West");
            add (d, "Center");
            add (num2, "East");
            add (result, "Center");
            add (outResult, "South");
            pack ();
            setResizable (false);
            setTitle ("Division");
            setVisible (true);
            d.addActionListener (this);
        }

        void actionPerformed (ActionEvent e) {
            if (e.getSource () == d) {
                int n1 = Integer.parseInt (num1.getText ());
                int n2 = Integer.parseInt (num2.getText ());
                out = String.valueOf (n1 / n2);
                outResult.setText (out);
            }
        }
    }
}

```

```

double resultNum;
int flag = 0;
public DivisionMain()
{
    setLayout(new FlowLayout());
    dResult = new JButton("RESULT");
    labelNumber1 = new JLabel("Number 1 : " + labelNum);
    labelNumber2 = new JLabel("Number 2 : " + labelNum);
    num1 = new JTextField(5);
    num2 = new JTextField(5);
    ourResult = new JLabel("Result : " + labelResult);
    add(labelNumber1);
    add(labelNumber2);
    add(labelResult);
    add(num1);
    add(num2);
    add(dResult);
    add(resultNum);
    num1.addActionListener(this);
    num2.addActionListener(this);
    dResult.addActionListener(this);
    addWindowListener(new WindowAdapter()
    {
        public void windowClosing(WindowEvent we)
        {
            System.exit(0);
        }
    });
    public void actionPerformed(ActionEvent ae)
    {
        double n1, n2;
        try
        {
            if(ae.getSource() == dResult)
            {
                n1 = Double.parseDouble(num1.getText());
                n2 = Double.parseDouble(num2.getText());
                num1.setText("");
                num2.setText("");
            }
            if(flag == 0)
            {
                out = n1 + " + " + n2;
                result = String.valueOf(out);
                repair();
            }
            else
            {
                g.drawString(out, 100, 100);
                out = "";
            }
        }
        catch(ArithmaticException e)
        {
            flag = 1;
            out = "Divide by 0 exception";
            repair();
        }
        catch(NumberFormatException e)
        {
            out = "Number format exception";
            repair();
        }
    }
}

```

Num:

/* if ($c_{n_2} == 0$)

throw new ArithmeticException(); */

-main()

(new JPanel());

newButton ("RESULT");

$r_1 = new JLabel ("Number 1:");$

$r_2 = new JLabel ("Number 2:");$

$t_1 = new JTextField ("5");$

$t_2 = new JTextField ("5");$

$TentField (t_1);$

$TentField (t_2);$

$newLabel ("Result:", labelResult);$

$newLabel ("Result:", labelResult);$

$label = n_1 + n_2;$

$result = String. valueOf (label);$

$repaint();$

Y

TentField (t_1);

TentField (t_2);

labelResult (labelResult);

0.0

Page

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

```
    DivisionName dm = new DivisionName();
    dm.setTitle ("New Division [800, 400]");
    dm.setVisible (true);
```

④ Testfield_Demo.java

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

```
public class TestfieldDemo extends Frame implements
```

```
ActionListener {
```

```
Testfield name; pane;
```

```
public Testfield Demo () {
```

```
setLayout (new FlowLayout());
```

```
label name = new Label ("Name:", Label.RIGHT);
```

```
label pass = new Label ("Password:", Label.RIGHT);
```

```
name = new TextField (12);
```

```
pass = new TextField (8);
```

```
pass.setEditable (true);
```

```
add (name);
```

```
add (pass);
```

```
add (pane);
```

```
pane.addActionListener (this);
add WindowListener (new MyWindowAdapter());
```

```
}
```

```
public void paint (Graphics g)
{ g.drawString ("Name: " + name.getText (), 100, 200);
g.drawString ("Selected test in name: " + name.getSelectedItem ());
}
```

⑤ Meaning ("parametrized class")

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

```
TestfieldDemo dm = new Testfield
```

```
dm.setTitle ("New Division (" +
```

```
dm.getWidth () + "x" + dm.getHeight () + ")");
```

```
dm.setVisible (true);
```

⑥ JButtonAdapter.java

```
class MyWindowAdapter extends Window
```

```
public void windowClosing (Window
```

```
System.exit (0);
```

```
}
```

⑦ JButtonAdapter.java

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

```
import java.util.ArrayList;
```

```
import javax.swing.Thread
```

```
import java.util.Random;
```

```
public class JButtonAdapter extends Frame
```

```
ActionListener {
```

```
int n;
int min;
```

```
Button[] buttons = new JButton[10];
String (label),
```

```
in cl;
```

void main (String args[])

```
dm = new Dimension(300, 200);
new Dimension(300, 200));
Dimension d1 = new Dimension("300, 200");
true);
```

```
g. creating ("password" + ran.nextInt(1, 100));
public static void main (String args[])
{
    TextFieldsDemo awin = new TextFieldsDemo();
    awin.setSize (new Dimension (300, 200));
    awin.setTitle ("TF_Label Demo");
    awin.setVisible (true);
```

```
Class MyWindowAdapter extends WindowAdapter
```

```
{ public void windowClosing (Window event we)
```

```
Id Demo extends Frame implements
```

```
par;
```

```
name1 =
```

```
w1.setFont (Font ("Name", Font.PLAIN);
```

```
new Label ("Name:", label1);
```

```
new Label ("Password:", label2);
```

```
label2.setFont (Font ("Name", Font.PLAIN);
```

```
label3.setFont (Font ("Name", Font.PLAIN);
```

```
label4.setFont (Font ("Name", Font.PLAIN);
```

```
label5.setFont (Font ("Name", Font.PLAIN);
```

```
label6.setFont (Font ("Name", Font.PLAIN);
```

```
label7.setFont (Font ("Name", Font.PLAIN);
```

```
label8.setFont (Font ("Name", Font.PLAIN);
```

```
graphis g)
```

```
; " + name.getText(), 100, 200);
gent in name; " + name.getText(),
Solen iste, 100, 200);
```

```
g. creating ("password" + ran.nextInt(1, 100));
public static void main (String args[])
{
    TextFieldsDemo awin = new TextFieldsDemo();
    awin.setSize (new Dimension (300, 200));
    awin.setTitle ("TF_Label Demo");
    awin.setVisible (true);
```

⑤ Buttondrag.java

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import java.util.Random;
```

```
import java.util.Random;
```

```
import java.awt.Button;
```

```
import java.awt.Window;
```

```
import java.awt.event.ActionListener;
```

```
new WindowAdapter();
```

```
new ActionListener() {
```

```
    public void actionPerformed (ActionEvent e)
    {
        int n = 3;
        int m = n * n;
```

```
        JButton button = new JButton ("");
        button.addActionListener (new WindowAdapter());
```

```
        button.addActionListener (new ActionListener());
```



```
else if (ac.getSource() == resourc && (doneFlag))
```

```
if (checkCount == 2)
```

```
{  
    repaint();  
    totalTime +=  
    deltaTime +  
    result.outcome(true);  
}
```

```
doneFlag = true;  
doneFlag = false;
```

```
doneFlag = true;
```

```
doneFlag = false;
```

```
for (int i=0; i<m; i++)  
{  
    if (ac.getSource[i] == b[i] && (!clicked))  
    {  
        b[i].setVisible(false);  
        clicked = b[i].getLabel();  
        ci = i;  
        Urneed = !Urneed;  
    }  
}
```

```
repaint();
```

```
b[ci].setLabel(b[ci].getLabel());
```

```
b[ci].setVisible(true);
```

```
b[ci].setLabel(" " + clicked);
```

```
Clicked = !Urneed;  
Urneed = Urneed;
```

```
public void paint(Graphics g)
```

```
if (doneFlag)
```

```
    setBackground(Color
```

```
Set Foreground (color
```

```
else
```

```
else
```

```
    setBackground (co
```

```
g.setBackground(new Co
```

```
g.drawString (newCo
```

```
    public void shuffle
```

```
    {  
        for (int i=0; i<
```

```
        Random number
```

```
        int num = nu
```

```
        swap (num, r);
```

```
        checkCount++;
```

```
}  
public void checkCorrect()
```

```
{  
    int checkCount = 0;
```

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

```
    if (b[i].getLabel().equals (String value[i]))
```

```
        checkCount++;
```

```

        ac.getResource() == Relation.ac(doneFlag))

        if (clock.count == 8)
            totalTime = totalTimeInSeconds();
        for (int i = 0; i < m; i++)
            b[i] = setVisible(false);
        doneFlag = true;
        reset();
        setEnabled(true);
        repaint();
        ready.setEnabled(false);
        msg = "Congratulations!, you finished the " + totalTime
              + " seconds !!";
        repaint();
    }

    public void paint(Graphics g)
    {
        if (doneFlag)
        {
            setBackground(Color.white);
            setForeground(Color.black);
            SetForeground(Color.white);
        }
        else
        {
            setBackground(Color.white);
            g.setFont(new Font("serif", Font.PLAIN, 24));
            g.drawString("Congratulation", 150, 250);
            public void shuffleStart()
            {
                for (int i = 0; i < m; i++)
                {
                    Random number = new Random();
                    int num = number.nextInt(9);
                    swap(num, i);
                }
            }
            label1.setText("String value of(" + i + ")");
        }
    }
}

```

```
public void reset()
```

```
{  
    for (int i = 1; i < m; i++)
```

```
        b[i].setVisible (true);
```

```
        b[i].setLabel (String.valueOf (i));
```

```
    }  
    b[0].setVisible (false);
```

```
    b[0].setLabel ("9");
```

```
    doneFlag = false;
```

```
    component[0] = com.getComponent();
```

```
    for (int a = 0; a < com.length; a++)
```

```
        com[a].setEnabled (false);
```

```
    reset. setEnabled (true);
```

```
    repaint();
```

```
}
```

```
public void reset ()
```

```
{  
    for (int i = 1; i < m; i++)
```

```
        b[i].setUsed (String.valueOf (i));
```

```
    }  
    b[0].setLabel ("9");
```

```
    component[0] = buttonPanel.getComponents();
```

```
    for (int a = 0; a < com.length; a++)
```

```
        com[a].setEnabled (false);
```

```
    public void swap (int x, int y)
```

```
{  
    String temp = b[x].getLabel();
```

```
    b[x].setLabel (b[y].getLabel());
```

```
    b[y].setLabel (temp);
```

```
}
```

public static
 JButton drag
 Cd - setsize
 ca. get title
 cd.setVisi

button drag
Cd - setsize
ca. get title
cd.setVisi

class Mywin
 public void
 {
 class Timea
 {
 private f

public void
 {
 long n
 run

public static void main(String args[])

{
button drag cd = new buttondrag();

cd - setsize (new Dimension (500, 500));

cd.setTitle ("Button Game");

cd.setVisible (true);

class MyWindowAdapter extends WindowAdapter

{
public void windowClosing (WindowEvent we)

{
System.exit(0);

{
}

class timecalc

{
}

private final long startedMillis = System.currentTimeMillis();

TimeMillis t;

public int getTimelineSeconds()

{
long nowMillis = System.currentTimeMillis();

return (int) ((nowMillis - startedMillis)/1000);

.getComponents();

{
},

{
},

Not

for

for