

18/12/22

write question

classmate

Date _____

Page _____

```
import java.util.Scanner;
import java.lang.Math;
class quadratic
{
    public static void main (String xx[])
    {
        System.out.println ("Mohammed Zeeshan Umar,
IBM22CS160");
        float a,b,c,d,r1,r2;
        Scanner sl = new Scanner (System.in);
        System.out.println ("enter the co-efficients");
        a = sl.nextFloat();
        b = sl.nextFloat();
        c = sl.nextFloat();
        d = b*b - (4*a*c);
        if (d > 0)
        {
            r1 = (-b + (float) Math.sqrt(d))/(2*a);
            r2 = (-b - (float) Math.sqrt(d))/(2*a);
            System.out.println ("r1 = " + r1);
            System.out.println ("r2 = " + r2);
        }
        else if (d == 0)
        {
            r1 = (-b)/(2*a);
            r2 = (-b)/(2*a);
            System.out.println ("r1 = " + r1);
            System.out.println ("r2 = " + r2);
        }
        else
        {
            r1 = (float) Math.sqrt(-d)/(2*a);
            r2 = -1 * r1;
        }
    }
}
```

```
System.out.println("root: " + "r1" + " = " + (-b) / (2 * a)) +  
    ("2 * a)) + " + "r1" + " i " + " \n r2 = " + (-b / (2 * a)) +  
    "r2" + " i ");
```

{

(1) $r_1 = \frac{-b}{2a}$ where $b < 0$ & $a > 0$

{

(2) $r_2 = \frac{-b}{2a}$ where $b > 0$ & $a < 0$

OutPut

Ques 2 Date 18/12/2022

(3) $r_1 = \frac{-b}{2a}$ where $b < 0$ & $a < 0$

(4) $r_1 = \frac{-b}{2a}$ & $r_2 = \frac{b}{2a}$ where $b > 0$ & $a > 0$

(5) $r_1 = \frac{-b}{2a}$ & $r_2 = -d$

(6) $r_1 = \frac{-b}{2a}$ & $r_2 = d$

(7) $r_1 = \frac{-b}{2a} - d = b$

(8) $b < h$

Ques 3

((a+b))((b) * p) * HtM((b+d)) + d = 18

((a+b))((b) * p) * HtM((b+d)) - d = 8r

((a+b) * ir) * HtM((b+d)) = 18

((a+b) * sr) * HtM((b+d)) = 8r

(a+b) is only

((a+b))((d+r) * 18)

((a+b))((d+r) * 8r)

((a+b) * 18) * HtM((b+d)) = 18

((a+b) * 8r) * HtM((b+d)) = 8r

LAB - 02 .

Q. A Java program to create a Student with members: usn, name, an array credits and an array marks. Include methods to accept and display details and method to display SGPA of students.

```

System.out.println("Enter student details");
System.out.print("Enter USN: ");
String usn = System.in.read();
System.out.print("Enter Name: ");
String name = System.in.read();
System.out.print("Enter Credits: ");
int credits[] = new int[8];
System.out.print("Enter Marks: ");
float marks[] = new float[8];

public class Sgpa {
    String usn, name;
    int credits[] = new int[8];
    float marks[] = new float[8];
    public void accept_name(String name, String usn) {
        this.name = name;
        this.usn = usn;
    }
    public void accept_credits(int credits[]) {
        this.credits = credits;
    }
    public void accept_marks(float marks[]) {
        this.marks = marks;
    }
    public String return_usn() {
        return this.usn;
    }
    public String return_name() {
        return this.name;
    }
    public int[] return_credits() {
        return this.credits;
    }
    public float[] return_marks() {
        return this.marks;
    }
}

```

```

1. Public float Sgpa ( int credits[], float marks[] )
2. int total_credits = 0;
3. float marks_credits = 0;
4. for ( int i = 0; i < 8; i++ ) {
5.     total_credits = total_credits + credits[i];
6.     marks_credits = marks_credits + (marks[i] * marks[i]);
7. }
8. float avg_marks = total_credits / marks_credits;
9. return marks_credits / total_credits;
}

```

```

public static void main (String [] args) {
    Sgpa p = new Sgpa ();
    p.accept_name ("Mohammed Zeeshan Umar");
    int [] credits_Brd = {4, 4, 4, 3, 3, 2, 1, 1};
    p.accept_credits (credits_Brd);
    float [] marks_Brd = {8.2344f, 7.544484f,
                          7.3456f, 8.2346f, 8.4677f, 8.3855f,
                          7.345346f, 7.5345f};
    p.accept_marks (marks_Brd);
    System.out.println ("User! " + p.return_avg());
    System.out.println ("Name! " + p.return_name());
    int [] gets_credits = p.return_credits ();
    for (int i = 0; i < 8; i++) {
        System.out.println ("marks: " + i + ", get: " + marks[i]);
    }
    System.out.println ();
    System.out.println ("Sgpa! " + p.sgpa (get_credits,
                                         get_marks));
}

```

{}

Output : "H" + "i" + "i" + "l" + "P" + "o" + "d" + "i"

VSN : IBM22CS160

Name : Mohammed Zeeshan Umar

Credits : 4 credits : 4 credits : 4 credits : 3 credits : 3
credits : 2 credits : 1 credits : 1

Marks : 8.23340 marks : 7.5445 marks : 7.345600

Marks : 8.2346 marks : 8.14677 marks : 8.3455

Marks : 7.3953 marks : 7.1534540.

Lgpa : 7.9169

(Grade fix 1.2 on point).

Q. A class Book which contains four members : name, author, price, num + pages. Include a constructor to set the values for the members. Include methods to set and get the details of the object. Include a "toString()" method that could display the complete details of the Book. Develop a Java program to create n book objects.

Sol

```
import java.util.Scanner; // import package  
class Book{ // defining the class  
    String name, author; int p;  
    int num + pages, price;  
    Book (String n, String a, int np, int p);  
        this.name = n;  
        this.author = a;  
        this.numpage = np;  
        this.price = p;  
  
    public String toString(){  
        return "Name: " + name + "Author" +
```

author + "Price" + price + "no.page" +
 num_page;
 }.

Class Book Det

Static Scanner & Scanner S = Scanner (System.)

Static Book set()

S.nextLine();

System.out.println("Enter book");

String n = S.nextLine();

System.out.println("author : ");

String a = S.nextLine();

System.out.println("Enter price of book");

int P = S.nextInt();

if (P < 0) {

System.out.println("Invalid price");

System.exit(1);

System.out.println("no. page : ");

int np = S.nextInt();

if (np < 0) {

System.out.println("Invalid");

System.exit(1);

Book bl = new Book (n,a,np,p);

return bl;

}

public static void main (String [] args) {

int n;

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

n = S.nextInt();

{ bl.setBook (n);

System.out.println("Author" + author + "Name" + name + "Price" + price + "no.page" + num_page);

```
Book b[ ] = new Book[n];  
for (int i=0; i<n; i++)  
    b[i] = new Book();  
System.out.println("Details of book entered");  
for (int i=0; i<n; i++)  
    System.out.println(b[i]);
```

* Develop a java program with the following:

Output:

Enter the no. of books

1

Enter book name

rd sharma

Enter author name

rd

Enter price

200

Enter no. of pages.

350

Details of books entered

Book name : rd sharma

Author : rd

Price : 200

No. of pages : 350.

By
Gaurav

8/11/2X

LAB - 3



- Q. Develop a java program with an abstract class named Shape that contain 2 integer and an empty method printArea. Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extend the class Shape. Each one of the classes contain only the method printArea of the given shape.

```
abstract class Shape {  
    abstract void printArea();  
    int a = 10;  
    int b = 2;  
}
```

```
class Rectangle extends Shape
```

```
    void printArea()  
    {
```

```
        System.out.println("area of rectangle is"  
                           + (a * b));  
    }
```

```
class Triangle extends Shape
```

```
    void printArea()  
    {
```

```
        System.out.println("area of triangle is"  
                           + (0.5 * a * b));  
    }
```

Class Circle extends shape

{ void printArea ()

{

{ System.out.println ("area of circle is " + (3.14 * a * a));

}

class Area

{

public static void main (String [] args)

{

Shape b = new Circle ();

b.printArea ();

b = new Rectangle ();

b.printArea ();

b = new Triangle ();

b.printArea ();

{

Output :-

area of circle is 314.0

~~area~~ of rectangle is 20

~~area~~ of triangle is 10.0.

By
Arun

22/1/24

LAB-4

classmate

Date _____
Page _____

Q. Create a package CIE which has two classes - Students and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of student. Create another package SEE which has the class External which is derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of student. Import the two packages in a file that declares the final marks of n students in all five courses.

Sof

Main Directory

- CIE
 - Student.java
 - Internals.java
- SEE
 - External.java
- CalculateMarks.java

(1) package CIE ;

```
public class Student {  
    String usn;  
    String name;  
    int sem;
```

```
Student (String usn, String name, int sem) {
```

```
this.usn = usn;  
this.name = name;  
this.sem = sem;  
}
```

```
}
```

(ii) package CIE;

```
public class Internal {  
    int[] marks;
```

```
    Internal(int[] marks) {
```

```
        this.marks = marks;
```

```
}
```

```
}
```

(iii) package SEE;

```
import CIE.Student;
```

```
public class External extends Student {
```

```
    int[] marks;
```

```
    marks);
```

```
    External(String usn, String name, int sem, int[]
```

```
    super(usn, name, sem);
```

```
    this.marks = marks;
```

```
3
```

```
3
```

(iv) import CIE.Student;

```
import SEE.External;
```

```
import CIE.Internal;
```

```
public class CalculateMarks {
    public static void main (String [] args) {
        int [] internalMarks = {80, 75, 85, 90, 70};
        InternalMarks internals = new InternalMarks(internalMarks);
        int [] externalMarks = {75, 70, 80, 85, 65};
        ExternalMarks external = new External ("IBM22CS160", "Zeehan");
        int [] finalMarks = new int [5];
        for (int i=0; i<5; i++) {
            finalMarks[i] = internals.marks[i] + external.marks[i];
        }
        for (int i=0; i<5; i++) {
            System.out.println("Final Marks for course " +
                (i+1) + ":" + finalMarks[i]);
        }
    }
}
```

Output :-

Final Marks for Student Zeehan:
29/11/20

Final Marks for Student Zeehan:
Course 1: 77.5
Course 2: 72.5
Course 3: 82.5
Course 4: 87.5
Course 5: 67.5

Q. Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age < 0. In Son class, implement a constructor that takes both father and son's age and throws an exception if son's age is \geq father's age.

Sol. import java.util.Scanner;

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

```
class Father {  
    int age;  
}
```

```
Father(int age) throws WrongAge {  
    if (age < 0) {  
        throw new WrongAge("Age cannot be negative");  
    }  
}
```

```
this.age = age;
```

```
int getAge() {  
    return age;  
}
```

Page

```
class Son extends Father {  
    int sonAge;  
    son(int fatherAge, int sonAge) throws WrongAge {  
        super(fatherAge);  
        if (sonAge >= fatherAge) {  
            throw new WrongAge("Son's age should be less than father's Age");  
        }  
        this.sonAge = sonAge;  
    }  
}
```

```
int getsonAge() {  
    return sonAge;  
}
```

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

```
try {  
    System.out.println("Enter Father's Age: ");  
    int fatherAge = scanner.nextInt();  
}
```

```
System.out.println("Enter Son's Age: ");  
int sonAge = scanner.nextInt();
```

```
Father father = new Father(fatherAge);  
System.out.println("father's Age: " + father.getAge());
```

```
Son son = new Son(fatherAge, sonAge);  
System.out.println("Son's Age: " + son.getSonAge());  
} catch (WrongAge e) {  
    System.out.println("Exception: " + e.getMessage());  
}
```

~~class WrongAge {
 public void checkAge() {
 if (age < 0) {
 System.out.println("Age cannot be negative");
 }
 }
}~~

Output:

→ Enter Father's Age : 45

Enter Son's Age : 20

Father's Age : 45

Son's Age : 20

→ Enter Father's Age : -10

Age cannot be negative

→ Enter Father's Age : 40

Enter Son's Age : 45

Son's age should be less than father's Age

Run
29/11/20

5/2/24

Date _____
Page _____
Page _____

→ Implement Basic program on thread.

class Thread1

{

public static void main(String ss[])

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

t.setName("Current main Thread");
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[Thread-5, Main].

CT: Thread[Current Main Thread, 5, main].

5

4

3

2

1

G2/V
Q. Write a program that creates two threads one thread displaying BMSCE in every 10 sec. sec and another displays CSF every 2 sec

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("BMS COLLEGE OF ENGG");
Thread.sleep(10000);
}
}

catch(InterruptedException ie)
{

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

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

}

```
class BMSThread
{
    public static void main (String ss[])
    {
        new NewThread();
        System.out.println("Back in main");
    }

    for (int n=10; n>0; n--)
    {
        System.out.println("CSE");
        Thread.sleep(2000);
    }

    catch (InterruptedException ie)
    {
        System.out.println("Main Thread interrupted");
    }

    System.out.println("Main Thread quitting");
}
```

Output:

C:\> Thread [NThread, 5, main].

Back in main

BMS College of ENGG

CSE

CSE

CSE

CSE

BMS COLLEGE OF ENGG

CSE

BMS COLLEGE OF ENGG

5/2/2024

19/2/24

LAB-8

Q. Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The saving account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-Acc and Sav-Acc to make them more specific to their requirements. Include the necessary methods in order to achieve the following task:

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

Check for the minimum balance, impose penalty if necessary and update the balance.

Sol. import java.util.Scanner;

```
class Account {
    String customerName;
    int accountNumber;
    double balance;
```

```
Account (String customerName, int accountNumber) {
```

```
    this.customerName = customerName;
```

```
    this.accountNumber = accountNumber;
```

```
    this.balance = 0.0;
```

```
}
```

```
void deposit (double amount) {
```

```
    balance += amount;
```

```
    System.out.println ("Balance: INR " + balance);
```

```
}
```

```
void displayBalance () {
```

```
    System.out.println ("Balance : INR " + balance);
```

```
}
```

```
void compoundInterest () {
```

```
    System.out.println ("Compound Interest not available");
```

```
}
```

```
void withdraw (double amount) {
```

```
    if (amount <= balance) {
```

```
        balance -= amount;
```

```
        System.out.println ("Withdrawal successful");
```

```
}
```

```
else {
```

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

```
}
```

```
class CurrentAccount extends Account {  
    double minBalance;  
    double serviceCharge;
```

```
CurrentAccount(String CustomerName, int accountNumber,  
    double minBalance, double serviceCharge) {  
    super(CustomerName, accountNumber);  
    this.minBalance = minBalance;  
    this.serviceCharge = serviceCharge;  
}
```

```
void withdraw(double amount) {  
    if(balance - amount >= minBalance) {  
        balance = balance - amount;  
        System.out.println("Withdrawal successful");  
    } else {  
        System.out.println("Insufficient balance. Service  
            charge of INR " + serviceCharge);  
        balance = balance - serviceCharge;  
    }  
}
```

```
class SavingAccount extends Account {  
    double interestRate;
```

~~```
SavingAccount(String CustomerName, int
 accountNumber, double interestRate) {
 super(CustomerName, accountNumber);
 this.interestRate = interestRate;
}
```~~

```
void compoundInterest() {
 double interest = balance * (interestRate / 100);
 balance = balance + interest;
 System.out.println("Interest of INR " + interest);
}
```

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

```
SavingsAccount savingAccount = new
 SavingsAccount("Mohammed Zeeshan Umar", 1000);
CurrentAccount currentAccount = new
 CurrentAccount("Mannendra Singh Chouhan", 1000, 1000);
```

```
System.out.println("Enter deposit amount for
 Saving account: ");
double savingDepositAmount = scanner.nextDouble();
SavingsAccount.deposit(savingDepositAmount);
SavingsAccount.displayBalance();
SavingsAccount.compoundInterest();
SavingsAccount.displayBalance();
```

```
System.out.println("Enter deposit amount of
 Current Account: ");
double currentDepositAmount = scanner.nextDouble();
CurrentAccount.deposit(currentDepositAmount);
CurrentAccount.displayBalance();
```

```

System.out.println("Enter withdrawal amount.");
double withdrawalAmount = scanner.nextDouble();
currentAccount.withdraw(withdrawalAmount);
currentAccount.withdrawAndDisplayBalance();

```

Scanner.close();

RJ

3.

Output :

Enter deposit amount for saving accounts : 60000  
Deposit successful.

Balance : INR 60000.

Interest : INR 3000.

Balance : INR 63000 .

Enter deposit amount of current account : 70000  
Deposit successful.

Balance INR : 70000.0.

Enter withdrawal amount : 35000.

withdrawal successful.

Balance : 35000.0.

Fm  
19/12/22

## LAB - 09 .

Q. Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num2 and Num1 is displayed in the Result field when the divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a `NoNumberFormatException`. If Num2 were zero, the program would throw an `ArithmeticException`. Display the exception in a message dialog box.

Sol)

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class IntegerDivisionUI extends JFrame {
 private JTextField num1Field, num2Field, resultField;

 public IntegerDivisionUI() {
 setTitle("Integer Division Calculator");
 setSize(300, 200);
 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 setLocationRelativeTo(null);

 JPanel panel = new JPanel();
 panel.setLayout(new GridLayout(4, 2));

 panel.add(new JLabel("Num1:"));
 num1Field = new JTextField();
 panel.add(num1Field);
```

```
panel.add(new JLabel("Num2:"));
num2Field = new JTextField();
panel.add(num2Field);
```

```
panel.add(new JLabel("Result:"));
resultField = new JTextField();
resultField.setEditable(false);
panel.add(resultField);
```

```
JButton divideButton = new JButton("Divide");
divideButton.addActionListener(new ActionListener() {
 public void actionPerformed(ActionEvent e) {
 try {
 int num1 = Integer.parseInt(num1Field.getText());
 int num2 = Integer.parseInt(num2Field.getText());
 int result = num1 / num2;
 resultField.setText(String.valueOf(result));
 } catch (NumberFormatException ex) {
 JOptionPane.showMessageDialog(null, "Please
 enter valid integers for Num1 and Num2.");
 } catch (ArithmaticException ex) {
 JOptionPane.showMessageDialog(null, "Cannot
 divide by zero.");
 }
 }
}); panel.add(divideButton);
add(panel);
setVisible(true);
```

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

Output: (num1) is odd and the result  
of (num1) and (num2) is 510.

Num1 : 12

Num2 : 6

Result: (2) is odd and the result  
of (num1) and (num2) is 510.

Divide: (1) is even and the result  
of (num1) and (num2) is 200.

Num1 : 12

Num2: One is not divisible by  
another number.

Result: (1) is not divisible by another  
number.

Divide: (1) is not divisible by zero.  
Not  
possible to divide by zero.

cannot divide by zero.

(1) is not divisible by another number.

(1) is not divisible by another number.