

B.M.S COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU



AAT

22CS3PCOOJ

LAB OBSERVATION

Submitted in partial fulfillment of the requirements for LAB COMPONENT

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Manvendra Singh Chouhan
(1BM22CS148)

Table Of Contents

Sl No	Title
1	Quadratic equations
2	Student SGPA Calculator
3	Book using <code>toString()</code>
4	Shape using abstract class
5	Bank (Savings/Current account)
6	Exception Handling (Father and Son)
7	Threads
8	Division of two integers on user interface
9	Packages (CIE / SEE)

Ques1. Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

Q Java program to find the solution of the following Quad equation.

Input -

```
import java.util.Scanner;  
  
public class Main {  
    public static void main (String[] xx) {  
        Scanner input = new Scanner (System.in);  
        System.out.println ("Enter the coeff of a : ");  
        double a = input.nextDouble();  
        System.out.println ("Enter the coeff of b : ");  
        double b = input.nextDouble();  
        System.out.println ("Enter the coeff of c : ");  
        double c = input.nextDouble();  
  
        double d = b*b - 4*a*c;  
        System.out.println ("Discriminant : " + d );  
        if (d > 0) {  
            double sol1 = (-b + sqrt (d ))/(2*a);  
            double sol2 = (-b - sqrt (d ))/(2*a);  
            System.out.println ("The Real solutions are : sol1 = " + sol1 + " & sol2 = "  
                + sol2);  
        }  
    }  
}
```

```

        else if (d == 0) {
            double sol = -b / (2 * a);
            System.out.println("One real solution is: Sol = " + sol);
        }
    } else {
        System.out.println("No real solution, discriminant is negative");
    }
}

```

Date: 1/1/24

(xv) Chapter 10: Linear Equations

(i) (a > 0) discriminant < 0

(ii) (a < 0) discriminant < 0

O/P?

Output: Enter the coefficient of a : 1
 Enter the coefficient of b : 2
 Enter the coefficient of c : 3
 Discriminant :-8

No real solution, discriminant is negative

$$a^2 b^2 + d^2 - 4ab \geq 0$$

$$(b + d)^2 \geq 0$$

$$(a < b)$$

$$(a+d)(b+d) \geq 0$$

$$(a+d)(b+d) \geq 0$$

(a > 0) $\Rightarrow a^2 b^2 + d^2 \geq 0$ (a < 0) $\Rightarrow a^2 b^2 + d^2 \geq 0$

Ques2. Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student

1.1.12-1

A program to create Student with members usn, name, an array credits and array marks. calculate SGPA.

```
import java.util.Scanner;
class Student {
    private String usn;
    private String name;
    int credits[] = new int[5];
    int marks[] = new int[5];
    void display(String usn, String name, int cr[], int marks[]) {
        System.out.println("USN: " + usn + " Name: " + name);
        for (int i = 0; i < 5; i++) {
            System.out.println("Credits: " + cr[i] + " Marks: " + marks[i]);
        }
    }
    void sgpa(int cr[], int marks[]) {
        int maxCreMarks = 0, creMarks = 0;
        for (int i = 0; i < 5; i++) {
            maxCreMarks += cr[i] * 10;
            creMarks += ((marks[i] / 10) + 1) * cr[i];
        }
        double sgpa = (double)(creMarks * 10) / maxCreMarks;
        System.out.println("SGPA: " + sgpa);
    }
    public static void main(String[] args) {
        String usn, name;
        int credits[] = new int[5];
        int marks[] = new int[5];
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter USN: ");
        usn = sc.nextLine();
    }
}
```

```

System.out.println("Enter Name : ");
name = s1.next();

System.out.println("Enter credits of all subjects : ");
for (int i=0; i<5; i++) {
    credits[i] = s1.nextInt();
}

System.out.println("Enter marks of all subjects : ");
for (int i=0; i<5; i++) {
    marks[i] = s1.nextInt();
}

```

Student student = new Student();
 Student.display(usn, name, credits, marks);
 Student.sgpa(credits, marks);
 }

Output :

Enter USN :

1BM22CS148

Enter Name : Manvendra

Enter credits for 5 subjects :

4 4 3 3 1

Enter the marks for all subjects :

91 96 88 86 93

USN : 1BM22CS148

Name : Manvendra

Credits : 4 Marks : 91

Credits : 4 Marks : 96

Credits : 3 Marks : 88

Credits : 3 Marks : 86

Credits : 1 Marks : 93

SGPA: 9.6

✓

Ques3. Create 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 objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

Java program to develop/create a class book which contains four members name, author, price, num_pages with constructor to set values for the members. include methods to set and get details of the object. include `toString()` method that could display the complete details of the book. Create n book objects.

```

import java.util.Scanner;
class Book {
    String name, author;
    int num_pages, price;
    Book(String n, String a, int np, int p) {
        this.name = n;
        this.author = a;
        this.num_pages = np;
        this.price = p;
    }
    public String toString() {
        return "Name = " + name + "\n Author = " + author + "\n Price = "
               + price + "\n no of pages = " + num_pages;
    }
}
class BookDet {
    static Scanner s = new Scanner(System.in);
    static Book get() {
        s.nextLine();
        System.out.println("Book name");
        String n = s.nextLine();
    }
}

```

```

System.out.println("Author Name");
String a = s.nextLine();
System.out.println("Price of the book");
int p = s.nextInt();
if (p < 0)
{
    System.out.println("Invalid Price");
    System.exit(1);
}
System.out.println("No. of pages");
int np = s.nextInt();
if (np < 0)
{
    System.out.println("Invalid Systemi((1)); Number of pages");
}
book b1 = new Book(n, a, np, p);
return b1;
}

public static void main (String sx[])
{
    int n = Integer.parseInt(sx[0]);
    System.out.println("Mobile no");
    System.out.println("Enter number of books");
    n = s.nextInt();
    Book b[] = new Book[n];
    for(int i=0; i<n; i++)
    {
        b[i] = new Book();
        b[i].set();
        System.out.println("Details of Book entered");
    }
    for(int i=0; i<n; i++)
    {
        System.out.println(b[i]);
    }
}

```

Output

option 161

Enter number of books

+

Enter book name

A

Enter Author name

A

Enter Price

100

Enter Pages

200

Invalid

Enter Price

-6

Invalid Price

R/
8/12/2021

} separate character separator exists

} () mismatch brace

} (((d+d)+c) * a) digit to char } string character type

} separate character separator separator exists

} () mismatch brace

} (((d+d)+c) * a) digit to char } string character type

} separate character separator exists

} () mismatch brace

} (((d+d)+c) * a) digit to char } string character type

Ques4. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method print Area() that prints the area of the given shape.

B | Jan | 24

Q Develop a java program with an abstract class named Shape that contains two integers and an empty method printArea(). Provides 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.

```
import java.awt.*;  
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 Rectangle 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)

public static void main (String xx []) {

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

R/
8/11/2020

Ques5. 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 savings 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_acct and Sav_acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: a) Accept deposit 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

LAB

Q Develop a java program to create a class bank that maintains two kinds of account for customers one - saving by other - current facility but no cheque book facility the current one provides cheque book but no interest. Current acc holder should also maintain a minimum balance if if the balance falls below this level, a service charge will be imposed.

Create class Account that stores customer name, account number as type of account. From this derive the class Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods for achieving following task:

- Accept deposit from customer to update the balance.
- Display current balance
- Compute and deposit interest
- Permit withdrawal to update the balance.

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

Code:

```

import java.util.Scanner;
class Account {
    String customerName;
    int accNumber;
    double balance;
    Account (String customerName, int accNumber) {
        this.customerName = customerName;
        this.accNumber = accNumber;
        this.balance = 0.0;
    }
    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited");
    }
}

```

```
Void DisplayBalance () {
```

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

```
Void compoundInterest() {
```

```
    System.out.println (" C.I. not Available ");
```

```
Void withdrawl (double amount) {
```

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

```
        balance = balance - amount;
```

```
        System.out.println (" Withdrawn ");
```

```
    } else {
```

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

```
}
```

```
}
```

```
Class CurrentAccount extends Account {
```

```
    double minBalance;
```

```
    double serviceCharge;
```

```
CurrentAccount (String customerName, int accNumber, double minBalance,  
double serviceCharge) {
```

```
    super (customerName, accNumber);
```

```
    this.minBalance = minBalance;
```

```
    this.serviceCharge = serviceCharge;
```

```
}
```

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

```
    if (balance - amount >= minBalance) {
```

```
        balance = balance - amount;
```

```
        System.out.println (" Its Withdrawn ");
```

```

else {
    System.out.println("Insufficient Balance. Service charge of INR " +
        balance = balance - serviceCharge; // serviceCharge );
}

class SavingAccount extends Account {
    double BiinterestRate;
    SavingsAccount (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);
        savingAccount Savings Account = new SavingAccount ("Mohammed Zeeshan
            Current Account");
        currentAccount Current Account = new CurrentAccount ("Manavendra Singh Chauhan");
        System.out.println ("Enter deposit amt for savings Account");
        double savingsDepositAmount = scanner.nextDouble();
    }
}

```

```

savings Account. deposit (savings DepositAmount);
savings Account. display Balance();
Savings Account. compound Interest();
savings Account. display Balance();
System.out.println("Enter deposit amount for current account:");
double current Deposit Amount = scanner.nextDouble();
current Account. deposit (current Deposit Amount);
current Account. display Balance();
System.out.println("Enter withdrawal amt");
double withdrawal Amount = scanner.nextDouble();
Current Account. withdraw (withdrawal Amount);
Current Account. display Balance();
Scanner.close();
}
}

```

Output:

Enter deposit amount for saving account: 60000

Deposit Successful

Balance INR 60000.0

Interest of INR 3000.0

Balance: INR 63000.0

Enter deposit amount for current account: 70000.

Deposit Successful

Balance INR 70000.0

Enter withdrawal amt 35000.0

Withdrawal Successful

Balance: INR 35000.0

Dr/
20/12/20

Ques6. 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 checks both father and son's age and throws an exception if son's age is >=father's age.

Q Work a program that demonstrates handling of exceptions in inheritance tree. Create base class "Father" and derived class called "Son" which extends 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 cases both father and son's age and throws an exception if son's age >= father's age

Input

```

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 -ive");
        }
        this.age = age;
    }
    int getAge () {
        return age;
    }
}

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() );
        }
    }
}
```

Output:-

→ Enter Father's Age : 45
Enter Son's Age : 19
Father's Age : 45
Son's Age : 19

→ Enter Father's Age : -45
Age cannot be negative -ive
→ Enter Father's Age : 40
Enter Son's Age : 60

Son's Age should be less than Father's Age.

Ques7. Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds

LAB.

Q) Create a program which creates 2 threads one displaying "BMS College of Engineering" once every 10 seconds. & another displaying "CSE" once every 2 seconds.

Code:

```
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("CSE: " + n);
                Thread.sleep(2000);
            }
        }
        catch(InterruptedException ie)
        {
            System.out.println("Child Thread Interrupted");
        }
        System.out.println("Child thread quit");
    }
}

class Thread2
{
    public static void main(String ss[])
    {
        System.out.println("Parent thread");
    }
}
```

```

new NewThread());
System.out.println("Back in Main");
try {
    for (int n = 8; n > 0, n--) {
        System.out.println("BMSCL: " + n);
        Thread.sleep(1000);
    }
} catch (InterruptedException ie) {
    System.out.println("Main thread " + ie);
}
System.out.println("Main thread exiting");

```

Output:

(CT : Thread#20, NTuscan, s, main)

Back in main

CSE: 15

BMSCE:3

CE 24

CSF:3

CSE 11

BASCET = 2

Childhood & Adolescent Quitting

BMSCF: 1

Main three

Ques8. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 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 NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box

Q) Java Program that creates interface to perform integer division
 User enters 2 numbers Num1 & Num2. The div of Num1 by Num-2 is displayed in the result when divide button is clicked.
 If NUM-1 or NUM-2 is not integer, the program would throw exception (NumberFormatException). If Num is zero then program throws ArithmeticException. Display the exception in message dialog box.

Code

```

import java.awt.*;
import java.applet.*;
import java.awt.event.*;
import javax.swing.*;

public class DivApplet extends Applet implements ActionListener {
    Label L1, L2, L3;
    JTextField T1, T2, Res;
    JButton B1;

    public void init() {
        L1 = new Label("Enter Num1:");
        add(L1);

        L2 = new Label("Enter Num2:");
        add(L2);

        T1 = new JTextField(10);
        add(T1);

        T2 = new JTextField(10);
        add(T2);

        B1 = new JButton("Divide");
        B1.addActionListener(this);
        add(B1);
    }

    public void actionPerformed(ActionEvent e) {
        String s1 = T1.getText();
        String s2 = T2.getText();

        try {
            int n1 = Integer.parseInt(s1);
            int n2 = Integer.parseInt(s2);
            int res = n1 / n2;
            Res.setText("Result = " + res);
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(this, "Please enter integer values");
        } catch (ArithmaticException ex) {
            JOptionPane.showMessageDialog(this, "Division by zero is not allowed");
        }
    }
}

```

```

LB = new Label("result");
add(LB);

Result
RF = new TextField(10)
add(RF);

BL = new Button("Divide");
add(BL);

BL.addActionListener(this); }

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == BL) {
        try {
            int value1 = Integer.parseInt(T1.getText());
            int value2 = Integer.parseInt(T2.getText());
            int res = value1 / value2;
            Result.setText(String.valueOf(res));
        } catch (NumberFormatException nfe) {
            JOptionPane.showMessageDialog(this, "Not a number");
        }
    }
    catch (ArithmeticException ae) {
        JOptionPane.showMessageDialog(this, "Divide by zero");
    }
}

```

Output

Applet

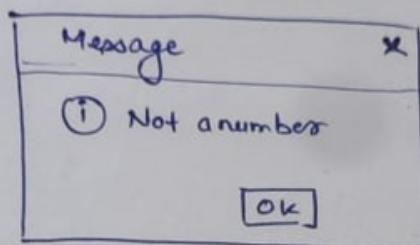
Enter Num1 : Enter Num2 : result

Divide

applet

Enter Num1 : Enter Num2 : result

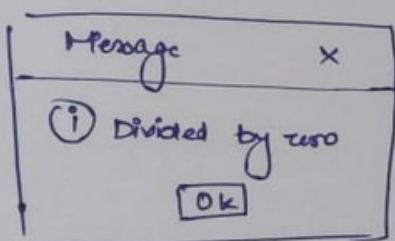
Divide



applet

Enter Num1 : Enter Num2 : result

Divide



Re:
20 / 20

Ques9. Create a package CIE which has two classes- Student 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 the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses

Input -

Main Directory

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

Q LNB - 22-3-24
Create a package CIE has two classes- Student and Internals. The class Student has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in 5 courses of the current sem of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in 5 courses of the current sem of student. Import 2 packages in a file that declares the final marks of n students in all 5 courses

(i) 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 Internals {
    int[] marks;
    Internals (int[] marks) {
        this.marks = marks;
    }
}
```

(iii) package SEE;

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

```

(iv) import CIE.student;
import SEE.External;
import CIE.Internals;

public class CalculateMarks {
    public static void main (String [] args) {
        int internalMarks = { 80, 75, 85, 90, 70 };
        Internals internal = new Internal (internalMarks);
        int [ ] internalMarks = { 75, 90, 80, 85, 65 };
        External external = new External ("1BM20CS148", "Manavendra",
                                         5, externalMarks);

        int [ ] finalMarks = new int [5];
        for (int i = 0, i < 5, i++) {
            finalMarks [i] = internalMarks [i] * 0.5 + externalMarks [i] * 0.5;
        }
        for (int i = 0, i < 5, i++) {
            System.out.println ("Final Marks for course " +
                                (i + 1) + ":" + finalMarks [i]);
        }
    }
}

```

Output:

Final Marks for student manavendra:
 Course 1 : 77.5
 Course 2 : 92.5
 Course 3 : 82.5
 Course 4 : 87.5
 Course 5 : 67.5

29/1/20

THANK YOU