

B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU



LAB REPORT

23CS3PCOOJ

Submitted in partial fulfilment of the requirements for Lab
Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

RAKESH K
1BM22CS216

Department of Computer Science and Engineering,
B.M.S College of Engineering,
Bull Temple Road, Basavanagudi, Bangalore, 560 019
2023-2024.

INDEX

SLNO	Title	Date
1	LAB 1	12/12/2023
2	LAB 2	19/12/2023
3	LAB 3	26/12/2023
4	LAB 4	02/01/2024
5	LAB 5	09/01/2024
6	LAB 6	16/01/2024
7	LAB 7	23/01/2024
8	LAB 8	30/01/2024
9	LAB 9	06/02/2024
10	LAB 10	20/02/2024
11	Complete Scanned Observation Book	12/12/2023- 20/02/2024

Lab Programs

1. 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.

```
import java.util.Scanner;
class quad {
    public static void main(String args[]) {
        int a, b, c;
        double r1, r2, d;
        Scanner s = new Scanner(System.in);
        System.out.println("Nehal A K\n1BM22CS176");
        System.out.println("Enter the coefficients of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
        while (a == 0) {
            System.out.println("Not a quadratic equation");
            System.out.println("Enter a non zero value for a:");
            a = s.nextInt();
        }
        d = b * b - 4 * a * c;
        if(d == 0) {
            r1 = (-b) / (2 * a);
            System.out.println("Roots are real and equal");
            System.out.println("Root1 = Root2 = " + r1);
        } else if(d > 0) {
            r1 = ((-b) + (Math.sqrt(d))) / (double) (2 * a);
            r2 = ((-b) - (Math.sqrt(d))) / (double) (2 * a);
            System.out.println("Roots are real and distinct");
            System.out.println("Root1 = " + r1 + " Root2 = " + r2);
        } else if(d < 0) {
            System.out.println("Roots are imaginary");
            r1 = (-b) / (2 * a);
            r2 = Math.sqrt(-d) / (2 * a);
```

```

        System.out.println("Root1 = " + r1 + " + i" + r2);
        System.out.println("Root1 = " + r1 + " - i" + r2);
    }

}

```

2. 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.

```

import java.util.Scanner;

class Subject {

    int subjectMarks;
    int credits;
    int grade;

}

```

```

class Student

{ String
    name;
    String usn;
    double SGPA;
    Subject[] subject;
    Scanner s;

```

```

Student() {
    int i;
    subject = new Subject[9];

```

```
    subject[i] = new Subject();
    s = new Scanner(System.in);
}

void getStudentDetails()
{
    System.out.println("Enter student
name"); name = s.next();
    System.out.println("Enter student usn");
    usn = s.next();
}
```

```
void getMarks() {
    int i;
    for (i = 0; i < 9; i++) {
        System.out.println("Enter marks for subject " + (i + 1));
        subject[i].subjectMarks = s.nextInt();
        System.out.println("Enter credits for subject " + (i + 1));
        subject[i].credits = s.nextInt();

        if (subject[i].subjectMarks >= 90) {
            subject[i].grade = 10;
        } else if (subject[i].subjectMarks >= 80 && subject[i].subjectMarks
< 90) {
            subject[i].grade = 9;
```

```

        } else if (subject[i].subjectMarks >= 70 && subject[i].subjectMarks
<80) {
            subject[i].grade = 8;
        } else if (subject[i].subjectMarks >= 60 && subject[i].subjectMarks
<70) {
            subject[i].grade = 7;
        } else if (subject[i].subjectMarks >= 50 && subject[i].subjectMarks
<60) {
            subject[i].grade = 6;
        } else if (subject[i].subjectMarks >= 40 && subject[i].subjectMarks
<50) {
            subject[i].grade = 5;
        } else
            { System.out.println("Failed"
            ); System.exit(0);
        }
    }
}

```

```

void computeSGPA()
{
    int totalCredits = 0;
    int creditsGained = 0;
    int i;

    for (i = 0; i < 9; i++) {
        totalCredits += subject[i].credits;
    }
}

```

```
    creditsGained += subject[i].credits * subject[i].grade;  
}
```

```
SGPA=(double) creditsGained / totalCredits;  
}
```

```
void displayResult()  
{ System.out.println("Name = " +  
name); System.out.println("Usn = " +  
usn); System.out.println("SGPA = " +  
SGPA);  
}
```

```
public class Main {  
    public static void main(String args[])  
    { Student s1 = new Student();  
        s1.getStudentDetails();  
        s1.getMarks();  
        s1.computeSGPA();  
        s1.displayResult();  
    }  
}
```

3. 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.

```
import java.util.Scanner;
```

```
class Book  
{ String  
    name; String  
    author; int  
    price;  
    int numPages;
```

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

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

```
    }

}

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

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

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for Book " + (i + 1) + ":");
            System.out.print("Name: ");
            String name = s.next();
            System.out.print("Author: ");
            String author = s.next();
            System.out.print("Price: ");
            int price = s.nextInt();
            System.out.print("Number of pages: ");
            int numPages = s.nextInt();
            books[i] = new Book(name, author, price, numPages);
        }
    }
}
```

```

System.out.println("\nDetails of the books:");
for (int i = 0; i < n; i++) {
    System.out.println("Book " + (i + 1) + ":\n" + books[i].toString());
}
}

```

4. 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 printArea() that prints the area of the given shape.

```

import java.util.*;

abstract class AbsArea {
    int a, b;

    AbsArea(int x)
    {
        a = x;
    }

    AbsArea(int x, int y)
    {
        a = x;
        b = y;
    }

    abstract void area();
}

class rec extends AbsArea {

```

```

rec(int a, int b) {
    super(a, b);
}

void area() {
    System.out.println("The area of the rectangle is: " + a * b);
}

class tri extends AbsArea {
    tri(int a, int b) {
        super(a, b);
    }

    void area() {
        System.out.println("The area of the triangle is: " + (a * b) / 2);
    }
}

class cir extends AbsArea {
    cir(int a) {
        super(a);
    }

    void area() {
        System.out.println("The area of the circle is: " + 3.14 * a * a);
    }
}

class Main {
    public static void main(String args[])
    { System.out.println("This is done by Nehal
AK\n1BM22CS176");
    int x, y;
    Scanner n = new Scanner(System.in);

    // Input for rectangle dimensions (x and y) with validation
}

```

```

x = n.nextInt();
y = n.nextInt();
if (x < 0 || y < 0) {
    System.out.println("Invalid input for rectangle. Please enter
positive values.");
    // You might want to handle this situation differently, such as
    asking the user to enter values again.
    System.exit(1); // Exiting with status code 1 (indicating abnormal
exit)
}

AbsArea r = new rec(x, y);
r.area();

// Input for triangle dimensions (x and y) with validation
System.out.println("Give input for triangle");
x = n.nextInt();
y = n.nextInt();
if (x < 0 || y < 0) {
    System.out.println("Invalid input for triangle. Please enter positive
values.");
    System.exit(1);
}

AbsArea t = new tri(x, y);
t.area();

// Input for circle radius (x) with validation
System.out.println("Give input for circle");
x = n.nextInt();
if (x < 0) {
    System.out.println("Invalid input for circle. Please enter a positive
value.");
    System.exit(1);
}

AbsArea c = new cir(x);

```

```
    c.area();
}
}
```

5. 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.

```
import java.util.Scanner;

class Account {
    String CustomerName;
    double AccNo, Balance;

    Account(String CustomerName, double AccNo, double Balance) {
        this.CustomerName = CustomerName;
        this.AccNo = AccNo;
        this.Balance = Balance;
    }

    public void Deposit(double Amount) {
        Balance += Amount;
        System.out.println("DEPOSIT SUCCESSFUL");
        DisplayBalance();
    }
}
```

```
void DisplayBalance() {  
    System.out.println("BALANCE:" + Balance);  
}  
}  
  
class CurrentAccount extends Account {  
    double MinBalance = 500.0;  
    double Charges = 10.0;  
  
    CurrentAccount(String CustomerName, double AccNo, double Balance) {  
        super(CustomerName, AccNo, Balance);  
    }  
  
    void Withdraw(double Amount) {  
        if (Balance >= Amount) {  
            Balance -= Amount;  
            System.out.println(Amount + " withdrawn successfully");  
            DisplayBalance();  
        } else {  
            System.out.println("insufficient Balance");  
        }  
    }  
  
    void UpdateBalance() {  
        if (Balance <= MinBalance) {  
            Balance -= Charges;  
        }  
    }  
}
```

```
        System.out.println("service charge applied for maintaining low
balance");

        DisplayBalance();

    }

}

}

class SavingsAccount extends Account {

    SavingsAccount(String CustomerName, double AccNo, double Balance) {
        super(CustomerName, AccNo, Balance);

    }

    double interest = 0.05;

    void UpdateBalance() {
        Balance = Balance + (interest * Balance);
        DisplayBalance();
    }

    void Withdraw(double Amount) {
        if (Balance >= Amount)
            { Balance -= Amount;
            DisplayBalance();
        } else {
            System.out.println("insufficient Balance");
        }
    }
}
```

```
}
```

```
class Bank {  
    public static void main(String args[]) {  
        String CustomerName;  
        double AccNo, Balance;  
        double amt, amt1;  
        Scanner in = new Scanner(System.in);  
  
        System.out.println("enter name:");  
        CustomerName = in.next();  
  
        System.out.println("enter AccNo:");  
        AccNo = in.nextDouble();  
        System.out.println("enter Balance:");  
        Balance = in.nextDouble();  
        System.out.println("enter amount to deposit");  
        amt = in.nextDouble();  
        System.out.println("enter amount to withdraw");  
        amt1 = in.nextDouble();  
  
        CurrentAccount c = new CurrentAccount(CustomerName, AccNo,  
        Balance);  
        c.Deposit(amt);  
        c.Withdraw(amt1);  
        c.UpdateBalance();
```

```

System.out.println(" ");
}

SavingsAccount s = new SavingsAccount(CustomerName, AccNo,
Balance);

s.Deposit(amt);

s.Withdraw(amt1);

s.UpdateBalance();

}

}

```

6.Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class Internals derived from Student 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.

```

package CIE;
public class Student
{
    public String name;
    public String usn;
    public int sem;

    public Student(String name,String usn,int sem)
    {
        this.name=name;
        this.usn=usn;
        this.sem=sem;
    }
}

```

```
package CIE;

public class Internals extends CIE.Student
{
    public int [] InternalMarks;

    public Internals(String name , String usn , int sem , int []InternalMarks)
    {
        super(name , usn , sem);
        this.InternalMarks=InternalMarks;
    }
}

package SEE;
import CIE.Student;

public class Externals extends Student
{
    public int [] SeeMarks;

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

import CIE.Student;
import CIE.Internals;
import SEE.Externals;
import java.util.Scanner;

public class FinalMarks
{
    public static void main(String [] args)
    {
        Scanner s1=new Scanner(System.in);

        System.out.println("Enter the number of Students");
        int n=s1.nextInt();
```

```

String []names=new String[n];
    String []usn=new String[n];
    int []sem = new int[n];
    int [][] InternalMarks = new int[n][5];
    int [][] SeeMarks = new int[n][5];

for(int i=0 ; i<n; i++)
{
    System.out.println("Enter details for Student" + (i+1) + ":");

    System.out.println("Name:");
    names[i]=s1.next();
    System.out.println("USN:");
    usn[i]=s1.next();
    System.out.println("SEM:");
    sem[i]=s1.nextInt();

    System.out.println("Enter Internal marks for 5 courses:");
    for(int j=0; j<5; j++)
    {
        System.out.println("Course"+(j+1)+":");
        InternalMarks[i][j]=s1.nextInt();
    }
    System.out.println("Enter External marks for 5 courses:");
    for(int j=0; j<5; j++)
    {
        System.out.println("Course"+(j+1)+":");
        SeeMarks[i][j]=s1.nextInt();
    }
}

int [][]FinalMarks = new int[n][5];
for(int i=0 ; i<n ; i++)
{
    Internals I1 = new Internals(names[i] , usn[i] , sem[i] , InternalMarks[i]);
    Externals E1 = new Externals(names[i] , usn[i] , sem[i] , SeeMarks[i]);

    for(int j=0; j<5 ;j++)
    {
        FinalMarks[i][j] = I1.InternalMarks[i] + E1.SeeMarks[j];
    }
    System.out.println("Finals Marks for " + n+ "Students in 5 courses:");
    for(i=0 ;i<n ;i++)
    {
        System.out.println(names[i] +":");
    }

    for(int j=0; j<5;j++)
    {
        System.out.println(FinalMarks[i][j] + ":");

    }
}

```

```

        }
        System.out.println();
    }
    s1.close();
}
}
}

```

7. 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 cases both father and son’s age and throws an exception if son’s age is >=father’s age.

```

import java.util.Scanner;

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

class InputScanner {
    static Scanner sc = new Scanner(System.in);
}

class Father extends InputScanner {
    int fatherAge;

    Father() throws WrongAge
    {
        System.out.println("Enter Father's
age"); fatherAge = sc.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
}

```

```
}

void display() {
    System.out.println("Father's age is " + fatherAge);
}

class Son extends Father {
    int sonAge;

    Son() throws WrongAge
    {
        System.out.println("Enter son's
age"); sonAge = sc.nextInt();
        if (sonAge > fatherAge) {
            throw new WrongAge("Son's age cannot be greater than father's age");
        } else if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

    void display() {
        System.out.println("Son's age is " + sonAge);
    }
}

public class ExceptionHandling {
    public static void Main(String args[]) {
        try{
            Son son= new Son();
            son.display();
        }
        catch(WrongAge e){
            System.out.println("Exception "+ e.getMessage());
        }
    }
}
```

1) class HelloWorld {

```
public static void main (String [] args) {  
    System.out.println ("Hello world");
```

{

o/p:

Hello world!

2) class RectangleArea {

```
public static void main (String [] args) {  
    int length, breadth;
```

```
length = Integer.parseInt (args [0]);
```

```
breadth = Integer.parseInt (args [1]);
```

```
int area = length * breadth;
```

```
System.out.print ("Length of rectangle = " + length);
```

```
System.out.print ("Length of rectangle = " + breadth);
```

```
System.out.print ("Area of rectangle = " + area);
```

{

{

o/p:

length = 10

breadth = 5

Area = 50.

a) import java.util.*;

class factorial

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

```
int fac = 1;
```

```
System.out.println ("Enter a number");
```

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

```
int n = S.nextInt ();
```

```
for (int i = 0, t = n; i < n; i++)
```

```
t = n;
```

```
for (int i = 0, fac = 1; i < n; i++)
```

```
System.out.println ("The factorial is " + fac);
```

obj.

Enter a number

The factorial is 120

5) class Array

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

```
int month[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 31 }
```

```
System.out.println ("April has " + month[3] + " days");
```

obj. April has 30 days

b) import java.util.*;

class palindrome

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

```
int n, t, rem, sum = 0;
```

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

```
System.out.println ("Enter a 5 digit number");
```

```
n = S.nextInt ();
```

```
t = n;
```

```
while (t > 0)
```

```
num = t % 10
```

```
sum = num * 10 + sum;
```

```
t = t / 10;
```

```
if (num == n)
```

```
System.out.println ("It is a palindrome");
```

```
else
```

```
System.out.println ("It is not a palindrome");
```

obj.

12321

it is a palindrome

LAB program-3

URBAN 19-12-22
EDG

Divide a Java program to create a class Student with members USN name, an array credits and an array marks, include methods to accept & display and a method to calculate GNP of a student.

```
import java.util.Scanner;

class Subject {
    int marks;
    int credits;
    String grade;
}

class Student {
    String name;
    String usn;
    double GPA;
    Scanner S;
    Subject Subject[7];
    Student() {
        int i;
        Subject = new Subject[7];
        for (i=0; i<9; i++) {
            Subject[i] = new Subject();
        }
        S = new Scanner(System.in);
    }

    void getMarks() {
        int i;
        for (i=0; i<9; i++) {
            System.out.println("Enter marks and credit");
            System.out.println("Course " + i + ":");
            int marks = S.nextInt();
            int credits = S.nextInt();
            Subject[i].marks = marks;
            Subject[i].credits = credits;
            if (marks >= 90 && marks <= 100)
                Subject[i].grade = "O";
            else if (marks >= 80 && marks < 90)
                Subject[i].grade = "A+";
            else if (marks >= 70 && marks < 80)
                Subject[i].grade = "A";
            else if (marks >= 60 && marks < 70)
                Subject[i].grade = "B";
            else
                Subject[i].grade = "C";
        }
    }

    void print() {
        System.out.println("Enter your USN:");
        System.out.println("Name:");
        System.out.println("Subject marks and credits");
        System.out.println("USN = " + S.nextLine());
    }
}
```

```
else if (marks >= 50 && marks < 60)
{
    Subject[i].grade = "B".
```

```
else if (marks >= 40 && marks < 50)
{
    Subject[i].grade = "C".
```

```
else if (marks >= 30 && marks < 40)
{
    Subject[i].grade = "D".
```

```
else if (marks >= 0 && marks < 30)
{
    Subject[i].grade = "F".
```

```
System.out.println ("The SGPA is : "+sgpa);
```

```
totalgradepoints += 5*Subject[i].credits;
```

```
case "F": totalgradepoints += 0*Subject[i].credits;
```

```
break;
```

```
case "C": totalgradepoints += 6*Subject[i].credits;
```

```
break;
```

```
case "B": totalgradepoints += 7*Subject[i].credits;
```

```
break;
```

```
case "B+": totalgradepoints += 7.5*Subject[i].credits;
```

```
break;
```

```
case "A": totalgradepoints += 8*Subject[i].credits;
```

```
break;
```

```
case "A+": totalgradepoints += 8.5*Subject[i].credits;
```

```
break;
```

```
Subject[i].grade = "P".
```

```
System.out.println ("The SGPA is : "+sgpa);
```

```
totalgradepoints / total credits;
```

```
double sgpa;
```

```
double totalgradepoints = 0;
```

```
double totalgradepoint = 0;
```

```
for (i = 0; i < 8; i++)
{
    total credits += Subject[i].credits;
}
```

```
Switch (Subject[i].grade)
```

```
(case "O": totalgradepoints += 10*Subject[i].
```

Output:-

```
Enter your name : Rakesh
```

```
Enter your USN : 216
```

```
Enter the marks and credits for course 0:
```

```
marks : 90
```

```
credits : 4
```

```
Enter the marks and credits for course 1:
```

```
marks : 92
```

```
credits : 4
```

URBAN 19/10/23

LAB PROGRAM - 3.

URBAN 26/10/23

entry the marks and credits for course 2
marks : 94
credit : 3

enter the marks and credits for course 3:
marks : 96
credit : 2

name : Rakesh
ISBN : 216
the price is : 10.00

Create a class Book which contains four
members : name, author, price, numPages. Include
a constructor to set the values for the
members. Include methods to set and get
the values 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.

```
import java.util.Scanner  
  
class Books  
{  
    String name;  
    String author;  
    int price;  
    int numPages;  
  
    Books (String name, String author, int price, int numPages)  
    {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = numPages;  
    }  
  
    public String toString()  
    {  
        String name, author, price, numPages;  
        name = "Book Name :" + this.name + "\n";  
        author = "Author Name :" + this.author + "\n";  
        price = "Price :" + this.price + "\n";  
        numPages = "Number of Pages :" + this.numPages + "\n";  
    }  
}
```

```
Return name + author + price + numPages;
```

b[i] : new books (name, author, price, numPages);

```
System.out.println ("Book Details :");  
or (i>0 ; i<n ; i++)
```

```
System.out.println (b[i]);
```

```
Scanner sc = new Scanner (System.in);  
int n;  
int i;
```

```
String name;
```

```
String author;
```

```
int price;
```

```
int numPages;
```

Output:

```
Enter the number of books: 2
```

```
Enter the details of book 2: abc
```

```
Enter the name of the book 1: abc
```

```
Enter the author name: xyz
```

```
Enter the price: 200
```

```
Enter the number of pages: 530
```

Books b[i]:

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

```
for (i=0 ; i<n ; i++)
```

```
System.out.println ("Enter the details of book " + (i+1));
```

```
System.out.println ("Enter the name of the book: ");
```

```
name = S.nextLine();
```

```
System.out.println ("Enter the author name: ");
```

```
author = S.nextLine();
```

```
System.out.println ("Enter the price: ");
```

```
price = S.nextInt();
```

```
System.out.println ("Enter the number of pages: ");
```

```
numPages = S.nextInt();
```

Price: 350
Number of pages: 500

Develop a Java program to create an abstract class named Shape

import java.util.Scanner;

class InputScanner

{ protected Scanner s;

public InputScanner()

{ s = new Scanner (System.in); }

public int getInput (String message)

{ System.out.println (message);

return s.nextInt();

}

abstract class Shape extends InputScanner

{ protected int a, b;

public Shape()

{ }

Shape s;

public void printArea()

{ }

}

class Rectangle extends Shape

{ protected int a, b;

public Rectangle()

{ Super(); }

public void printArea()

{ a = getInput ("Enter the length :");

b = getInput ("Enter the breadth :");

int area = a*b;

System.out.println ("Area of Rectangle = " + area);

}

class Triangle extends Shape

{ protected int a, b;

public Triangle()

{ }

Super();

public void printArea()

{ a = getInput ("Enter the side1 :");

b = getInput ("Enter the side2 :");

double area = 0.5 * a*b;

System.out.println ("Area of the triangle = " + area);

}

```
class circle extends Shape
```

```
{  
protected int r;  
public circle()
```

```
{  
super();  
}
```

```
public void printArea()
```

```
a = getinput ("Enter the radius : ");  
double area = 3.14 * a *  
System.out.println ("Area of the circle : " + a)
```

```
public class MainShape
```

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

```
Rectangle r = new Rectangle ();  
Triangle t = new Triangle ();  
Circle c = new Circle ();
```

```
r.printArea();  
t.printArea();  
c.printArea();
```

Output:

Enter the length : 3
Enter the breadth : 4

Area of the Rectangle : 12
Enter the side : 5
Enter the side : 6

Area of triangle : 15.0
Enter the radius : 4

Area of the circle : 50.24
~~50.24~~ 50.24

Develop a Java program to create a class Bank with current account and savings account.

```

import java.util.Scanner;
class Account {
    String name;
    int accno;
    String type;
    double balance;
    Account (String name, int accno, String type,
             double balance) {
        this.name = name;
        this.accno = accno;
        this.type = type;
        this.balance = balance;
    }
    void deposit (double amount) {
        balance += amount;
    }
    void withdraw (double amount) {
        if ((balance - amount) >= 0)
            balance -= amount;
        else
            System.out.println ("Insufficient balance");
    }
}

```

import java.util.Scanner;

```

class Current extends Account {
    private double minBal = 500;
    private double serviceCharge = 50;
    Current (String name, int accno, double balance) {
        super (name, accno, "current", balance);
    }
    void checkmin() {
        if (balance < minBal)
            System.out.println ("Balance is less than min
balance, Service charge is imposed");
    }
}

```

EDG 09.11.24

```
balance = ServiceCharges;"  
System.out.println ("Balance is :" + balance);  
  
}  
  
}  
  
class AccountMain  
{  
    public static void main (String args)  
    {  
        Scanner s = new Scanner (System.in);  
        System.out.println ("Enter the name :");  
        String name = s.nextLine();  
        System.out.println ("Enter the type (current / savings) :");  
        String type = s.nextLine();  
        System.out.println ("Enter the initial balance :");  
        int bal = s.nextInt();  
        System.out.println ("Enter the initial balance :");  
        double balance = s.nextDouble();  
  
        int ch;  
        double amount, amount2;  
        Account acc = new Account (name, accno, type,  
            balance);  
        switch (ch = new Scanner (System.in).nextInt())  
        {  
            case 1:  
                if (acc.type.equals ("Savings"))  
                    System.out.println ("Enter the amount :");  
                amount = s.nextInt();  
                acc.deposit (amount);  
                break;  
            case 2:  
                System.out.println ("Enter the amount :");  
                amount2 = s.nextInt();  
                acc.withdraw (amount2);  
                System.out.println ("Amount withdrawn :");  
                System.out.println (acc.balance);  
                break;  
            case 3:  
                System.out.println ("Enter the amount :");  
                amount2 = s.nextInt();  
                acc.withdraw (amount2);  
                System.out.println ("Amount withdrawn :");  
                System.out.println (acc.balance);  
                break;  
            case 4:  
                System.out.println ("Enter the choice :");  
                ch = s.nextInt();  
                switch (ch){  
                    case 1:  
                        System.out.println ("Enter the amount :");  
                        amount = s.nextInt();  
                        acc.withdraw (amount);  
                        System.out.println ("Amount withdrawn :");  
                        System.out.println (acc.balance);  
                        break;  
                    case 2:  
                        System.out.println ("Enter the amount :");  
                        amount2 = s.nextInt();  
                        acc.withdraw (amount2);  
                        System.out.println ("Amount withdrawn :");  
                        System.out.println (acc.balance);  
                        break;  
                }  
            default:  
                System.out.println ("Enter the amount :");  
                amount = s.nextInt();  
                acc.withdraw (amount);  
                System.out.println ("Amount withdrawn :");  
                System.out.println (acc.balance);  
                break;  
        }  
    }  
}
```

EDG 09.11.24

if (ch == 1):
 System.out.println ("Enter the amount :");
 amount1 = s.nextInt()
 acc.deposit (amount1);
 break;

if (ch == 2):
 System.out.println ("Enter the amount :");
 amount2 = s.nextInt()
 acc.withdraw (amount2);
 System.out.println ("Amount withdrawn :");
 System.out.println (acc.balance);
 break;

case 2: System.out.println ("Enter the amount :");
amount2 = s.nextInt()
acc.withdraw (amount2);
System.out.println ("Amount withdrawn :");
System.out.println (acc.balance);
break;

Question 1] Output -1-

```
type 1 : Bmsce
type 2 : Bmse
type 3 : Bmsec
type 4 : ms
type 5 : abcd
```

Case 3: (a. display(),
break;

case 4: System.out(0);

Output:

→ Enter the name:

Shrey

→ enter the type (current/savings) :-

current

→ enter the account number :

1001

→ enter the initial balance :

2000

→ welcome !

1. deposit 2. withdraw 3. display

1

choice Use answer 1000

→ menu :

1. deposit 2. withdraw 3. display

2

enter the amount : 500

→ menu :

1. deposit 2. withdraw 3. display

3

→ name : Shrey account type : current . below is the

Question 2] Length of s1 = 5

concatenation of s1 and s2 : rmse Bmsce

Question 3] to String() + 10

Ques

Question 4] The given string is " welcome to bmsce college ". The size begin, sizeEnd, and at begin values are 11, 16

The value of character assig: [r, m, s, c, e, -].

Question 5] 65 66 67 68 69 70
R M S C E

Question 6] Bmsce equals Bmse → true

Bmse equals college → False

Bmse equals ignore case BMSCE → true

Question 7] grubsting matched

s1 = "Bmse collage",

s2 = "welcome to Bmse collage & Engineering";

Question 8] true
False

Question 9] false
true

Question 10] Hello equal Hello → True
Hello = Hello → False

Question 11] The names in alphabetical order are:
apple
ball
cat
water

Question 12] sorted numbers [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Question 13] The

Question 14] Hello world

Question 15] scanner

public void displayStudentDetails() {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter USN : ");
usn = scanner.nextInt();
System.out.print("Enter name : ");
name = scanner.next();
System.out.print("Enter semester : ");
sem = scanner.nextInt();
}

4

{

public void inputStudentDetails() {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter USN : ");
usn = scanner.nextInt();
System.out.print("Enter name : ");
name = scanner.next();
System.out.print("Enter semester : ");
sem = scanner.nextInt();
}

5
6
7
8
9
10

11
12
13
14
15

16
17
18
19
20

21
22
23
24
25

26
27
28
29
30

31
32
33
34
35

36
37
38
39
40

41
42
43
44
45

46
47
48
49
50

51
52
53
54
55

56
57
58
59
60

61
62
63
64
65

66
67
68
69
70

71
72
73
74
75

76
77
78
79
80

81
82
83
84
85

86
87
88
89
90

91
92
93
94
95

96
97
98
99
100

101
102
103
104
105

106
107
108
109
110

```
//Internal.java
package CIE;
```

```
import java.util.Scanner;
```

```
public class Internal extends Student
```

```
protected int marks[] = new int[5];
```

```
public void inputInternal()
```

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

```
System.out.println("Enter Internal Marks for "+name);
```

```
for (int i = 0; i < 5; i++)
```

```
{
```

```
marks[i] = new int[5];
```

```
finalMarks = new int[5];
```

```
System.out.println("Enter the "+(i+1)+" internal mark");
```

```
marks[i] = Scanner.nextInt();
```

```
System.out.println("Enter "+marks[i]);
```

```
finalMarks[i] = marks[i] / 2 + super.marks[i];
```

```
System.out.println("Subject "+(i+1)+" : "+finalMarks[i]);
```

```
marks[i] = Scanner.nextInt();
```

```
finalMarks[i] = marks[i] / 2 + super.marks[i];
```

```
System.out.println("Subject "+(i+1)+" : "+finalMarks[i]);
```

```
marks[i] = Scanner.nextInt();
```

```
finalMarks[i] = marks[i] / 2 + super.marks[i];
```

```
System.out.println("Subject "+(i+1)+" : "+finalMarks[i]);
```

```
marks[i] = Scanner.nextInt();
```

```
}
```

```
//External.java
package SEE;
```

```
import CIE.Internal;
```

```
public class External extends Internal
```

```
protected int marks[];
```

```
protected int finalMarks[];
```

```
public External()
```

```
marks = new int[5];
```

```
finalMarks = new int[5];
```

```
System.out.println("Input SEE marks");
```

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

```
System.out.println("Enter SEE marks for "+name);
```

```
for (int i = 0; i < 5; i++)
```

```
{
```

```
marks[i] = new int[5];
```

```
finalMarks[i] = marks[i] / 2 + super.marks[i];
```

```
System.out.println("Subject "+(i+1)+" : "+finalMarks[i]);
```

```
marks[i] = Scanner.nextInt();
```

```
finalMarks[i] = marks[i] / 2 + super.marks[i];
```

```
System.out.println("Subject "+(i+1)+" : "+finalMarks[i]);
```

```
marks[i] = Scanner.nextInt();
```

```
}
```

Output:

Enter USN: 216

Enter name: RK

Enter Sem: 43

Enter 1st marks : 45

Subject 2 marks : 46

Subject 3 marks : 47

Subject 4 marks : 48

Subject 5 marks : 60

Subject 1 marks : 90

Subject 2 marks : 92

Subject 3 marks : 95

Subject 4 marks : 98

Subject 5 marks : 100

Enter USN: 218 Enter name: Kalki Enter sem: 4.

Enter 1st marks : 98

Subject 2 marks : 47

Subject 3 marks : 48

Subject 4 marks : 49

Subject 5 marks : 50

Subject 1 marks : 95

Subject 2 marks : 97

Subject 3 marks : 96

Subject 4 marks : 95

Subject 5 marks : 100

input SEE Extrawork

class Main

public static void main (String args)

int numofStudents = 2;

Extrawork final Marks[] = new Extrawork [numofStudents];

for (int i=0; i < numofStudents; i++)

finalMarks[i] = new Extrawork();

finalMarks[i].inputStudentDetails();

System.out.println ("Enter 1st marks: ");

finalMarks[i].input1stMarks();

System.out.println ("Enter 2nd marks: ");

finalMarks[i].input2ndMarks();

System.out.println ("Enter 3rd marks: ");

finalMarks[i].input3rdMarks();

System.out.println ("Enter 4th marks: ");

finalMarks[i].input4thMarks();

System.out.println ("Enter 5th marks: ");

finalMarks[i].input5thMarks();

System.out.println ("Displaying data : ");

for (int i=0; i < numofStudents; i++)

finalMarks[i].calcuFinalMarks();

finalMarks[0].displayFinalMarks();

finalMarks[1].displayFinalMarks();

USN : 210 name : RK Semester : 3

Subject 1 : 90

Subject 2 : 92

Subject 3 : 94

Subject 4 : 92

Subject 5 : 100

import java.util.Scanner;

class WrongAge extends Exception {

public WrongAge() {

super ("Age Error");

}

class InputScanner {

protected Scanner Scanner;

public InputScanner() {

Scanner = new Scanner (System.in);

public int nextInt() {

return Scanner.nextInt();

}

class Father extends InputScanner {

protected int fatherAge;

public Father() throws WrongAge {

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

if (fatherAge < 0) {

Scanner.nextLine ();

System.out.println ("Age cannot be negative");

public void display() {

System.out.println ("Father's age: " + fatherAge);

}

EDG3 30.11.24

EDG3 30.11.24

class Son extends Father {
private int sonAge;

public Son() throws WrongAge {

super();

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

SonAge = Super.nextInt();

if (SonAge >= fatherAge) {

throw new WrongAge("Son's age

cannot be greater than or equal

to father's age);

} else if (SonAge < 0) {

throw new WrongAge("Age cannot be

negative");

8

30.11.24

public void display() {

super.display();

System.out.println("Son's age: " + SonAge);

public class EmployeeHandlingDemo {
public static void main(String[] args) {

try {

Son Son = new Son();

} catch (WrongAge e) {

System.out.println("Error: " + e.getMessage());

WORK-8

WAP which creates two threads, one thread displaying "BMS College of Engineering" once every two seconds and another displaying "CSE" once every two seconds.

Class Bms Thread extends Thread

② override

public void run()

while (true) {

System.out.println ("BMS College of Engineering")

try {

Thread.sleep (10000);

catch (InterruptedException e)

e.printStackTrace();

CSE

public void run()

while (true) {

System.out.println ("CSE")

try {

Thread.sleep (10000);

catch (InterruptedException e)

e.printStackTrace();

BMS Thread extends Thread

② override

public void run()

System.out.println ("BMS College of Engineering")

try {

Thread.sleep (10000);

catch (InterruptedException e)

e.printStackTrace();

Output:

BMS College of Engineering

CSE

66 / 20 / 24

Lab work - 10A

EDG3 13.102.24

Demonstrate how producer communication and deadlock

```

class Producer implements Runnable {
    int q;
    public void run() {
        Thread t = new Thread(this, "Producer");
        t.start();
        synchronized (q) {
            while (!q.get()) {
                try {
                    System.out.println("In Consumer waiting\n");
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    System.out.println("InterruptedException caught");
                }
            }
            System.out.println("In Producer producing\n");
            notify();
        }
    }
}

class Consumer implements Runnable {
    int q;
    public void run() {
        Thread t = new Thread(this, "Consumer");
        t.start();
        synchronized (q) {
            while (!q.get()) {
                try {
                    System.out.println("In Producer waiting\n");
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    System.out.println("InterruptedException caught");
                }
            }
            System.out.println("In Consumer getting\n");
            q.put(1);
        }
    }
}

```

4

//Deadline

```

class Refined {
    public static void main (String args[]) {
        q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println ("All control to stop.");
    }
}

class A {
    synchronized void foo (B b) {
        String name = Thread.currentThread().getname();
        System.out.println (name + " entered A-foo");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println (name + " trying to call B.left()");
        }
        b.left();
    }
}

class B {
    synchronized void left() {
        System.out.println ("Inside A.left");
    }
}

class Consumer {
    int count = 0;
    public void run() {
        while (true) {
            if (count > 10) {
                break;
            }
            synchronized (this) {
                count++;
                System.out.println ("Consumer: " + count);
            }
        }
    }
}

class Producer {
    int count = 0;
    public void run() {
        while (true) {
            if (count > 10) {
                break;
            }
            synchronized (this) {
                count++;
                System.out.println ("Producer: " + count);
            }
        }
    }
}

```

URBAN 13.02.24
EDG3

URBAN 13.02.24
EDG3

```
void b() {
    System.out.println("Inside A.b()");
}

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();

    Deadlock() {
        Thread t = new Thread(this, "MainThread");
        t.start();
        a.foo(b);
    }

    public void run() {
        System.out.println("Back in main thread");
        b.b();
    }
}

public void run() {
    System.out.println("Back in other thread");
    a.foo(b);
}
```

↳ ↴ ↳ ↴

o/b-
Main Thread entered A.foo
Racing Thread entered B.b-
Main Thread trying to call B.b()
Inside A.back
Back in main thread
Racing Thread trying to call A.back()
Inside A.back
Back in other thread.

Lab week - 9

URGENT 20.02.24
EDGEE

```

wap Mat creates a user interface
to perform integer divisions.

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

User UserInterface {
    UserInterface() {
        JFrame jfrm = new JFrame("Divide App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlabs = new JLabel("Enter the dividend
        and divisor :");
        JTextField jtfs = new JTextField(8);

        JButton button = new JButton("Calculate");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                try {
                    int a = Integer.parseInt(jtfs.getText());
                    int b = Integer.parseInt(jlabs.getText());
                    int ans = a / b;
                    jlabs.setText("a = " + a);
                    jlabs.setText("b = " + b);
                    jlabs.setText("ans = " + ans);
                } catch (NumberFormatException e) {
                    jlabs.setText("Enter only integers!");
                }
            }
        });

        jfrm.add(jlabs);
        jfrm.add(jtfs);
        jfrm.add(button);
        jfrm.add(button);
    }
}

```

```

ActionListener t = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        System.out.println("Action went from a
        button field");
    }
};
jlabs.addActionListener(t);
jlabs.addActionListener(t);

```

```

ActionListener t = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(jtfs.getText());
            int b = Integer.parseInt(jlabs.getText());
            int ans = a / b;
            jlabs.setText("a = " + a);
            jlabs.setText("b = " + b);
            jlabs.setText("ans = " + ans);
        } catch (NumberFormatException e) {
            jlabs.setText("Enter only integers!");
        }
    }
};
jlabs.addActionListener(t);
jlabs.addActionListener(t);

```

URGENT

```
ifrm.setVisible(true);
```

```
public static void main (String args[]) {  
    SwingUtilities.invokeLater (new Runnable) {  
        public void run () {  
            new UserInterface();  
        }  
    };  
}
```

Output:

Enter the divisor and dividend:

A = 6 B = 2 Ans = 3

6
2
3
0
0
0
0