

I N D E X

Name Adikya Shaema Std _____ Sec A
(BH22CS01)
Roll No. _____ Subject _____ School/College Tata L A B

Sl. No.	Date	Title	Page No.	Teacher Sign/ Remarks
1	09/10/2024	Implementation of Quadratic Equation	5	HPR
2	16/10/2024	Student data consisting (roll, name, an array of credits and marks and method to calculate SGPA of each student)	7	HPR
3.	23/10/2024	Class Book data consisting (name, author, price, num_pages. Include methods to set and get the details of the objects display complete details of the book	12	HPR
4.	23/10/2024	Abstract class with shapes	15	HPR
5.	13/11/2024	Bank that maintains three kinds of account for customer, one called Saving account and Current Account	18	HPR
6.	13/11/2024	package - CIF and SEE	23	HPR
7.	20/11/2024	Exception handling with inheritance tree	28	HPR
8.	27/11/2024	Threads: Display BMS and CSE	31	HPR
9.	27/11/2024	User Interface for integer division	33	HPR
10.	27/11/2024	Interprocess Communication and deadlock	35	HPR

1) WAP to print "Hello world" in C++
(class HelloWorld)

```
public static void main(string args)
{
    system.out.println("Hello World");
}
```

2) WAP to check if a number is prime or not

```
public static void main(string args)
{
    int no = 4;
    if (no == 1 || no == 2)
    {
        else in system.out.println("Number is prime");
    }
    else if (no % 2 == 0)
    {
        System.out.println("Number is not prime");
    }
}
```

Output
number is not prime

Output

number is not prime

"Great input" output

WAP to print Fibonacci series
(
↳ called well)

public class Fibonacci

{

public static void main (String [] args)

{

int a = 0;

int b = 1;

int m = 10;

System.out.println (a + b);

for (int i = 2; i < m; i++)

{

 int sum = a + b;

 System.out.println (", " + sum);

 a = b;

 b = sum;

}

}

Output

0, 1, 1, 2, 3, 5, 8, 13, 21, 34

WAP to check if a triangle is scalene,
isosceles or equilateral

public class triangle

{

public static void main (String [] args)

{

int side a = 10

int side b = 10

int side c = 20

if (side a == side b || side b == side c)

{

 System.out.println ("Equilateral");

}

```

if (side a == side b || side b == side c || side c == side a)

```

```

    {
        System.out.println("isosceles");
    }

```

```

else
{
}

```

Output
Isosceles

5) WAP to calculate simple interest

```
public class interest
```

```
{
}
```

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

```
{
}
```

```
int P = 40000
```

```
int T = 3
```

```
int R = 7
```

```
int inter;
```

```
inter = P * T * R / 100;
```

```
System.out.println("Your interest is " + inter);
```

```
{
}
```

Output

Your interest is 84000

6) W.P. - to swap two numbers) fin
(a+b==) ab+a

public class SwapWithTemp {

 public static void main(String[] args)

{

 int a = 5;

 int b = 10;

 System.out.println("Before swapping: a=" +
 ", b=" + b);

 int temp = a;

 a = b;

 b = temp;

 System.out.println("After swapping: a=" +
 ", b=" + b);

}
3. In this program statements at 9th U (c)

Output

Before swapping: a=5, b=10

After swapping: a=10, b=5

✓ 2002P = 9 thi
MP = 5 thi
F = 9 thi

lastri thi

(001)*T*9 = ahi

"lastri" is lastri and "9" is 9th line of output. ahi

✓ 2002P is lastri

2002P is lastri

WAP to print 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;
public class QuadraticEquationSolver {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Name: Aditya Sharma");
    System.out.println("USN: IBM22CS021");
    System.out.print("Enter coefficient a: ");
    double a = scanner.nextDouble();
    System.out.print("Enter coefficient b: ");
    double b = scanner.nextDouble();
    System.out.print("Enter coefficient c: ");
    double c = scanner.nextDouble();
    double discriminant = b * b - 4 * a * c;
    if (discriminant > 0) {
        double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
        double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
        System.out.println("The equation has two real solutions: " + root1 + " and " + root2);
    } else if (discriminant == 0) {
        double root = -b / (2 * a);
        System.out.println("The equation has one real solution: " + root);
    } else {
        System.out.println("There are no real solutions");
    }
}

```

Output at last the trap at 9AM

Name: Sharmistha interchanging
USN: 1BN22CS021 trap at 9AM

Enter Coefficient a = 5 - terminal

Enter Coefficient b = 10 initial speed

Enter Coefficient c = 2 initial speed

The equation has two real solutions:

-6.2254033307085166 and -1.7749666926188

~~but we take only the first real solution~~

(initial) terminal = a decimal

(initial) initial = small otherwise the answer

: ("a" trap) initial = 10 initial

: ("c" trap) initial = a decimal

: ("b" trap) initial = a decimal

: ("d" trap) initial = a decimal

: ("c" trap) initial = 10 initial

: ("d" trap) initial = a decimal

: "a" * 5 + "d" * 10 = terminal equal

: ("a" + "d") * 10 = terminal equal

: ("a" * 5) (terminal) = 50 - start + d -) = trap equal

: ("a" * 5) (trap) = start + d -) = trap equal

and initial velocity at "C" trap the method

: ("c" trap + "d" * 10 = 10 initial = start)

s

: ("c" = terminal) if possible

: ("a" * 5) \ d = trap equal

Does our initial velocity at "C" initial = trap equal?

: ("trap" + "initial")

s

b. a. 0

: ("initial" trap or 10 initial) = trap equal

8
Revise a java program to create a class student with members name, marks, an array credits or 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;

void calculategrade() {

if (Subjectmarks >= 90) {
grade = 10;

} else if (Subjectmarks >= 80) {
grade = 9;

} else if (Subjectmarks >= 70) {
grade = 8;

} else if (Subjectmarks >= 60) {
grade = 7;

} else if (Subjectmarks >= 50) {
grade = 6;

} else if (Subjectmarks >= 40) {
grade = 5;

} else if (Subjectmarks >= 30) {
grade = 4;

} else if (Subjectmarks >= 20) {
grade = 3;

} else {
grade = 0;

}

}

class student {

String name;

String leon;

double SGPA;

Subject [] : subject ;
Scanner s = new Scanner (System.in);
Student s .
Subjects = new Subjects [8] ;
for (int i = 0 ; i < 8 ; i++)
Subjects [i] = new Subject ();

Void getstudentDetails () {
System.out.println (" Enter student name : ");
name = s.nextLine ();
System.out.println (" Enter student USN : ");
usn = s.nextLine ();
}

Void getmarks () {
for (int i = 0 ; i < 8 ; i++) {
System.out.println (" Enter marks for Subject " +
+ (i + 1) + " : ");
System.out.println ();
Subject [i].subjectMarks = s.nextInt ();
while ((Subject [i].subjectMarks > 100) ||
(Subject [i].subjectMarks < 0)) {
System.out.println (" Invalid marks ! Enter a
Value between 0 and 100 : ");
Subject [i].subjectMarks = s.nextInt ();
}

System.out.println (" Enter credits for Subject " +
(i + 1) + " : ");
Subject [i].credits = s.nextInt ();
Subject [i].calculateGrade ();
}

Void ComputeSGPA () {

```
int effectiveness = 0;  
int totalcredits = 0;  
for (int i = 0; i < 5; i++) {  
    effectiveness += subjects[i].grade *  
        subjects[i].credits;  
    totalcredits += subjects[i].credits;  
}
```

$$\text{SGPA} = (\text{double}) \frac{\text{effectiveness}}{\text{totalcredits}}$$

```
void displayDetails () {  
    System.out.println ("The Student Name: " + name);  
    System.out.println ("The Student USN: " + usn);  
    System.out.println ("SGPA: " + SGPA);  
}
```

```
public static void main (String [] args) {  
    Student student = new Student ();  
    student.getStudentDetails ();  
    student.getMarks ();  
    student.computeSGPA ();  
    student.displayDetails ();  
}
```

Output

obj.

Object of class

Object of class

Object of class

Object of class

Enter student name: aditya
Enter student USN: 1bm22cs021
Enter marks for subject 1: 98

Enter credits for subject 1: 4

Enter marks for subject 2: 80

Enter credits for subject 2: 3

Enter marks for subject 3: 55

Enter credits for subject 3: 2

Enter marks for subject 4: 60

Enter credits for subject 4: 4

Enter marks for subject 5: 55

Enter credits for subject 5: 3

Enter marks for subject 6: 60

Enter credits for subject 6: 4

Enter marks for subject 7: 65

Enter credits for subject 7: 4

Enter marks for subject 8: 85

Enter credits for subject 8: 3

Student Name: aditya

E-mail:

Student USN: IBM22CS021 all o other (1)

SGPA: 7.814814814815m: medium

at sup with few of variations) a

in question it about student. medium for

such a type of be listed off

itself here) last bottom (apart from

that's place off be listed elsewhere)

. higher stand in there) at. except

for example. like my traps.

E-mail me:

new prints

other prints

any time

organization the

using this address prints your prints) and

of course,

new new, all

other prints with

organization - very well

g

Please what prints will

if "P" + man, with "man" written

"and brother, with "man brother"

"a" + wife, with "wife"

"I" + mother, with "I" + mother"

g

(about well) will

(I, I am prints) man first with will

not enough) second man is remark

(secondly, when with "I" + mother)

Week-3

1) Create a class book which contains four members : name, author, price, numPages. In 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 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 = numPages;
```

```
}
```

```
    public String toString() {
```

```
        return "Book name:" + this.name + "\n" +
```

```
        "Author name:" + this.author + "\n" +
```

```
        "Price:" + this.price + "\n" +
```

```
        "Number of pages:" + this.numPages + "\n";
```

```
}
```

3

public class books {

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

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

```
        System.out.print ("Enter the number of books:");
```

```
int m = s.nextInt();
Book[i] books = new Book[m];
for (int i=0; i<m; i++) {
    s.nextLine();
    System.out.print("Enter name of the book:");
    String name = s.nextLine();
    System.out.print("Enter author of the book:");
    String author = s.nextLine();
    System.out.print("Enter the price of the book:");
    int price = s.nextInt();
    System.out.print("Enter the number of pages of
the book:");
    int numPages = s.nextInt();
    books[i] = new Book(name, author, price, numPages);
}
System.out.println("In Book Details:");
for (int i=0; i<m; i++) {
    System.out.println(books[i].toString());
}
s.close();
}
```

Output

```
Enter the number of books: 2
Enter name of the book: java programming
Enter author of the book: jhon
Enter the price of the book: 600
Enter the number of pages of the book: 350
Enter name of the book: c programming
Enter author of the book: grom
Enter the price of the book: 450
Enter the number of pages of the book: 300
```

८

4

Book details: (C) + T + m.2 = m tri
Book name: java programming ad E2 book
Author name: shon > i can tri > eaf
price: 600 (C) miltan.2

Book name: C programming
Author's name: Tom White
Price: 45.00
Number of pages: 300

~~(6) *Triturus vulgaris* tri-~~
~~richus~~ ~~vulgaris~~ ~~trichophrys~~ - [F.] Island

("Want a hand?") ~~Only two ways~~
Possibilities (at least) left
~~and the third)~~ ~~Only two ways~~

~~Scallop~~ go ~~when~~ at ~~not~~

Digitized by srujanika@gmail.com

and each of the sections of

~~as stated at the meeting~~

774: *Acacia* sp. var. *varia* (Benth.) Benth.

ANSWER: The total cost of the 100 shirts is \$1000.

2) 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 contains only the method getArea() that prints the area of the given shape.

```
import java.util.Scanner;
class InputScanner {
    Scanner scanner = new Scanner(System.in);
    public int readInt(String prompt) {
        System.out.print(prompt);
        return scanner.nextInt();
    }
}
```

3

abstract Class shape extends InputScanner {

- int dimension1;
- int dimension2;
- Shape (int dimension1, int dimension2) {
 this.dimension1 = dimension1;
 this.dimension2 = dimension2;
 }

abstract void printArea();

class Rectangle extends shape {
 Rectangle (int length, int breadth) {
 super (length, breadth);
 }
}

void printArea () {
 int area = dimension1 * dimension2;
 System.out.println("Area of Rectangle : " + area);
}

3

21
class triangle extends Shape {

triangle (int base, int height) {

super (base, height);

}

void printarea() {

double area = 0.5 * dimension1 * dimension2;

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

}

3
class circle extends Shape {

circle (int radius) {

super (radius, 0);

}

void printarea() {

double area = Math.PI * dimension1 * dimension2;

System.out.println("Area of circle:" + area);

}

3
public class shapes {

public static void main (String args) {

InputScanner inputScanner = new InputScanner();

int length = inputScanner.readInt ("Enter length of Rectangle");

int breadth = inputScanner.readInt ("Enter breadth of Rectangle");

rectangle rectangle = new Rectangle (length, breadth);

rectangle.printarea();

int base = inputScanner.readInt ("Enter base of Triangle");

int height = inputScanner.readInt ("Enter height of Triangle");

triangle triangle = new triangle (base, height);

triangle.printarea();

int radius = inputScanner.nextInt("Enter radius of circle");

Circle circle = new Circle(radius);

circle.printArea(); // prints area of circle

better way, overload it if there

new circle at 300 toward original
output very large original at first

Enter length of Rectangle: 50 (inches)

Enter breadth of Rectangle: 80 (inches)

Area of Rectangle: 2000 (inches²)

Enter base of triangle: 30 (inches)

Enter height of triangle: 20 (inches)

Enter radius of circle: 45 (inches)

Area of Circle: 6361.725123819332 (inches²)

HP

normal, left, right, trapizoid

bottom, top

innermost point

outermost point

general point

leftmost point

the point from which) toward right

! (left point)

innermost point

outermost point

leftmost point

rightmost point

b (trapezoid which shaped like right

triangle).fi

c (trapezoid like)

F1

WGG T-5

LAB "Program - 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.

~~Ques~~

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;

String accountnumber;

double balance;

String accounttype;

public Account (String name, String accno,
String type) {

customername = name;

accountnumber = accno;

accounttype = type;

balance = 0.0;

}

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

System.out.println("Deposited: " + amount);

3

else

2

System.out.println("Deposit amount must be positive.");

3

public void displayBalance() {

System.out.println("Balance in account " + accountNumber + ":" + balance);

3

public void withdraw(double amount) {

if (amount <= balance) if (amount > 0) {

balance -= amount;

System.out.println("Withdrawn: " + amount);

3

else

System.out.println("Insufficient balance or invalid amount.");

3

class Saver extends Account {

public Saver (String name, String accNo) {

super (name, accNo, "Savings");

3

public void computeAndDepositInterest(double rate, int time) {

double interest = balance * Math.pow(1 + rate, 100 * time) - balance;

balance += interest;

System.out.println("Interest is "+interest+" deposited to your account.");

3

3

class Current extends Account {
 double minBalance; //
 double servicecharge;
 public Current (String name, String accno,
 double minBalance) {
 super (name, accno, "Current");
 minBalance = minBalance;
 servicecharge = 50.0;

3

public void checkMinimumBalance() {
 if (balance < minBalance) {
 balance += servicecharge;
 System.out.println("Balance is below
 minimum required. Service charge of "+servicecharge
 + " imposed.");

3

public class Bank {

public static void main (String [] args) {
 Scanner sc = new Scanner (System.in);
 Account account; //
 System.out.print ("Enter customer Name");
 String name = sc.nextLine();
 System.out.print ("Enter account Number: ");
 String accno = sc.nextLine();
 System.out.print ("Enter account type (1 for
 savings, 2 for current): ");
 int acctype = sc.nextInt();

```

if ( acctype == 1 ) {
    account = new CurrentAccount( name, accno );
} else if ( acctype == 2 ) {
    System.out.print("Enter minimum balance for current account:");
    double minbalance = sc.nextDouble();
    account = new CurrentAccount( name, accno, minbalance );
} else {
    System.out.println("Invalid account type. Exiting ...");
}

int option;
do {
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display Balance");
    if ( account instanceof Savings ) {
        System.out.println("4. Compute and Deposit Interest");
    }
    System.out.println("5. Exit");
    System.out.print("choose an option:");
    option = sc.nextInt();
    switch (option) {
        case 1:
            System.out.print("Enter deposit amount:");
            double depositamount = sc.nextDouble();
            account.deposit( depositamount );
            break;
    }
}

```

Case 2 -

double system exit point ("Enter withdrawl
amount");
double withdrawal amount = sc. acceptable;
Recent criticism (withdrawl amount);
breakdown - overlapping several

Case 3: environmental issues

account.displayBalance();

~~breast~~ ~~is~~ ~~in~~ ~~the~~ ~~middle~~ ~~of~~ ~~the~~ ~~body~~

Case 4:

```
if (count instances of Subject) >  
System.out.print("Enter average interest  
rate:");
```

double rate = < . nextDouble();

```
System.out.print("Enter time in years:");
```

`int time = sc.nextInt();`

(1) (a) (i) (strict) Account : Compute and deposit interest
(rate, time);

Else if 1 second instance of context 1

(C correct) orient 1. Check Minicompliance();

8

Dress;

Case 5 - ~~As I have been taught~~
System.out.println("Exiting... thank you for using
our banking services.");

Dredge; ("Fig. 7") ordinary dredge

~~Default is <code>no wait</code> every two seconds~~

System-out.println ("Invoked option. Very again");
break;

3

3 while(~~char~~ != '\n'); // read a line

-73

Cleistocactus *reniformis* *nana* *reinhardii* *sp. nov.*

Bonaire
Dolomit 1 =

1. deposit
2. withdraw
3. balance

3 - ~~alpha~~ Salome

3- ~~upper~~
4- complete set

4. Computer or expert
5. Back

8-8000
Enter recent annual WOB
as required & type in 8-8000

WEEK - 6

LAB program 6

package CIE;

import java.util.Scanner;

```
public class Student {
    protected String USN;
    protected String name;
    protected int sem;
```

```
public void inputStudentDetails() {
    Scanner s = new Scanner(System.in);
    System.out.print("Enter USN:");
    this.USN = s.nextLine();
    System.out.print("Enter Name:");
    this.name = s.nextLine();
    System.out.print("Enter Semester:");
    this.sem = s.nextInt();
```

3

```
public void displayStudentDetails() {
    System.out.println("USN:" + USN);
    System.out.println("Name:" + name);
    System.out.println("Semester:" + sem);
```

3

package CIE;

import java.util.Scanner;

public class Internals extends Student {

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

public void inputCIMarks() {

Scanner s = new Scanner(System.in);

System.out.println("Enter internal Marks for
5 subjects");

for (int i = 0; i < 5; i++) {
 System.out.print("Enter marks for
 Subject " + (i + 1) + ": ");
 this.marks[i] = s.nextInt();

{}

3

public void displayIEmarks() {
 System.out.println("Internal marks:");
 System.out.println("Subject " + (i + 1) + " : " + marks[i]);
 for (int i = 0; i < 5; i++) {
 System.out.println("Subject " + (i + 1)
 + " : " + marks[i]);

{}

3

{}

package SEE;
import CIF.External;
import java.util.Scanner;
public class External extends External {
protected int[] externalMarks = new int[5];
protected int[] finalMarks = new int[5];
public External() {
externalMarks = new int[5];
finalMarks = new int[5];

{}

public void calculateFinalMarks() {

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

finalMarks[i] = marks[i] + externalMarks[i];

{}

public void displayFinalMarks() {

displayStudentDetails();

displayIEmarks();

```

System.out.println("External marks:");
for (int i=0; i<t; i++) {
    System.out.print("Subject " + i + " External Marks: ");
    int externalMarks = scanner.nextInt();
    externalMarks[i] = externalMarks;
}

System.out.println("Selfied" + (i+1) + ":" +
    externalMarks[i]);
}

System.out.println("Final Marks:");
for (int i=0; i<t; i++) {
    System.out.print("Subject " + (i+1) + ":" +
        finalMarks[i]);
}

// calculate the average and
// total of all subjects
int sum = 0;
int count = 0;
for (int i=0; i<t; i++) {
    sum += finalMarks[i];
    count++;
}
double average = sum / count;

// calculate the total marks
int totalMarks = 0;
for (int i=0; i<t; i++) {
    totalMarks += finalMarks[i];
}

// calculate the percentage
double percentage = (totalMarks / 500) * 100;

// calculate the grade
String grade;
if (percentage >= 90) {
    grade = "A";
} else if (percentage >= 80) {
    grade = "B";
} else if (percentage >= 70) {
    grade = "C";
} else if (percentage >= 60) {
    grade = "D";
} else {
    grade = "E";
}

System.out.println("Total Marks: " + totalMarks);
System.out.println("Percentage: " + percentage);
System.out.println("Grade: " + grade);

// calculate the final marks
for (int i=0; i<t; i++) {
    finalMarks[i] = calculateFinalMarks(externalMarks[i], internalMarks[i], average);
}

System.out.println("Final Marks for all Student:");
for (int i=0; i<n; i++) {
    System.out.println("Student " + (i+1) + " Final Marks: " + finalMarks[i]);
}

```

System.out.println("Student" + (i+1) + ")\nStudents[i].displayInternalMarks();

3

Output

javac -d C:\Student.java C:\Internal.java

javac -d C:\External.java

java Main.java

Enter number of students : 1

Enter details for student 1

Enter USN : 1bm22cs021

Enter name : aditya sharma

Enter Semester = 3

Enter Internal Marks for 5 subjects :

Enter marks for Subject 1 : 35

Enter marks for Subject 2 : 37

Enter marks for Subject 3 : 25

Enter marks for Subject 4 : 30

Enter marks for Subject 5 : 31

Enter External marks for 5 subjects :

Enter marks for Subject 1 : 95

Enter marks for Subject 2 : 85

Enter marks for Subject 3 : 90

Enter marks for Subject 4 : 80

Enter marks for Subject 5 : 96

USN : 1bm22cs021

Name : aditya sharma

Semester : 3

Internal Marks :

Subject 1 : 35

Subject 3 : 37 at 1000 ms. 90%
Subject 3 : 20

Subject 2 : 4 = 30
Subject 2 : 5 = 31

Subject 1 : 130 (normal) + SFE

Subject 2 : 122 (normal) + SFE

Subject 3 : 115 (normal) + SFE

Subject 4 : 110 (normal) + SFE

Subject 5 : 127 (normal) + SFE

15

Q&P that demonstrates handing of exception in inheritance tree create a base class called "Father" and a derived class called "Son" which extends the base class. In Father's class implement a constructor which takes the age and throws the exception WrongAge() when the input age is less than zero. In Son's class implement a constructor that uses Father and Son's age and throws an exception if Son's age is greater than or equal to father's age.

import java.util.Scanner;

Class WrongAgeException extends Exception {
public WrongAgeException (String message);
super (message);

{

Class SonAgeException extends Exception {
public SonAgeException (String message);
super (message);

{

Class Father {

private int age;

public Father (int age) throws WrongAgeException
if (age < 0) {

throw new WrongAgeException ("Wrong!");

{

this.age = age;

{

public int getAge () {

return age;

{

{

```

class Son extends Father {
    private int sonAge;
    public Son (int fatherAge, int sonAge) throws WrongAgeException, SonAgeException {
        Super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException ("Son's age cannot be greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge () {
        return sonAge;
    }
}

public class FatherSon {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print ("Enter Father's Age:");
        int fatherAge = sc.nextInt();
        System.out.print ("Enter son's Age:");
        int sonAge = sc.nextInt();
        Son son = new Son (fatherAge, sonAge);
        System.out.println ("Accepted Successfully!");
    }
}

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

```

PS

System.out.println("Would you like to
re-enter details? Y/N") ;
String input = scan.nextLine();
if (input.equalsIgnoreCase("N")) {
break;
}

3

Output

Enter Father's Age: 40

Enter Son's Age: 20

Accepted successfully
Would you like to re-enter details? Y/N

Enter Father's Age: 40

Enter Son's Age: 50

Son's age cannot be greater than or equal
to Father's age

Would you like to re-enter details? Y/N

27/11/24 Lab program 8

- Q W&P which creates two threads, one thread displaying "BMS college of Engineering" every ten seconds and another displaying "CSE" once every two seconds.

Class BMS extends Thread {

 public void run() {

 try {

 while (true) {

 System.out.println("BMS college of engineering");

 Thread.sleep(10000);

 }

 } catch (InterruptedException e) {

 }

}

Class CSE extends Thread {

 public void run() {

 try {

 while (true) {

 System.out.println("CSE");

 Thread.sleep(2000);

 }

 } catch (InterruptedException e) {

 }

}

public class multithreading {

 public static void main(String[] args) {

 BMS bms = new BMS();

 CSE cse = new CSE();

 bms.start();

 cse.start();

}

}

~~Output~~

2 major part (a) & (b)

BMS College of Engineering

CSE merit seat category wise 9th &

CEG is called 2nd year principal merit

CSE sections has marks not grades

CSE course merit rank was "3rd"

BMS College of Engineering

CSE placed students 2nd & was

CSE (most bad setup)

CSE best

CSE best

BMS College of Engineering

CSG : local place board

CSE

CEG (2nd merit category wise) was 8

CSF

CSF

BMS College of Engineering

CLC : more less setup

CSE

placed 9th

(3rd merit category wise)
was 9th

CEG (2nd merit category wise) was 8

8

Engineering college and value

Engineering institution wise value

(LMR was 8 and 2nd)

(PSU was 5th in 82)

(Tatyasaheb)

Lab program 9

WAP creates a user interface to perform integer division. 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.

Output:

Enter divisor and dividend:
100 50 calculate button
 $A = 100$
 $B = 50$
 $Ans = 2$

Enter the divisor and dividend:

10.0 0 calculate button

~~B Should be non-zero~~

LAB-10

Demonstrate inter process communication and deadlock with respect to the implementation of two producers and consumer. (will do small to avoid
deadlock all nodes being itself act as
main, will do small if both is added)

Output report add negative as for
Reading thread entered B.bar & except
Main thread entered A.fas
Reading thread trying to call B.last()
Reading thread trying to call A.last()

Inside A.last

Bar in other thread

Inside B.last

Bar in main thread

Producer Waiting

Get: 11

Consumed: 11 has arrived

Put: 12

Intimate Consumer

Producer Waiting

get: 12

consumed: 12

put: 13

Intimate Consumer

get: 13

consumed: 13

put: 14

Intercrate consumer

got: 14

Consumed 14

18
got 14

10/09/2023

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
import java.awt.BorderLayout;  
  
class SwingDemo1  
{  
    SwingDemo1()  
    {  
        JFrame jfrm = new JFrame("Calculator App");  
        jfrm.setSize(275, 150);  
        jfrm.setLayout(new FlowLayout());  
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        JLabel jlab = new JLabel("Enter the divisor and dividend:");  
        JTextField jtlf = new JTextField(8);  
        JTextField jtlf2 = new JTextField(8);  
        JButton button = new JButton("calculate");  
        JLabel err = new JLabel();  
        JLabel slab = new JLabel();  
        jfrm.add(jlab);  
        jfrm.add(jtlf);  
        jfrm.add(jtlf2);  
        jfrm.add(button);  
        jfrm.add(err);  
        jfrm.add(slab);  
        button.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent evt) {  
                try {  
                    int a = Integer.parseInt(jtlf.getText());  
                    int b = Integer.parseInt(jtlf2.getText());  
                    int ans = a / b;  
                    slab.setText("A = " + ans);  
                } catch (Exception e) {}  
            }  
        });  
    }  
}
```

```
blab.setText ("B = "+b);
anslab.setText ("Ans = "+ans);
err.setText ("");
} catch (NumberFormatException e) {
    blab.setText ("");
    anslab.setText ("");
    err.setText ("Enter only integers!");
} catch (ArithmaticException e) {
    blab.setText ("");
    anslab.setText ("");
    err.setText ("B should be NON-zero!");
}
```

```
} );
System.setVisible (true);
}

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

Lab prog 10

class Q {

int m;

boolean Valuset = false;

Synchronized int get() {

while (! Valuset) ;

try {

System.out.println("In Consumer Waiting");

Cexit();

} catch (InterruptedException e) {

System.out.println("InterruptedException caught");

}

}

System.out.println("Cat: " + m);

Valuset = false;

System.out.println("In Intimate Producer(m)");

Notify();

return m;

}

Synchronized void put(int m) {

while (Valuset) ;

try {

System.out.println("In Producer Waiting(m)");

Cexit();

} catch (InterruptedException e) {

System.out.println("InterruptedException caught");

}

}

this.m = m;

Valuset = true;

System.out.println("Put: " + m);

System.out.println("In Intimate consumer(m)");

Notify();

}

}

Class Producer implements Runnable

Q q;

Producer(Q q){

this.q=q;

new Thread(this, "producer").start();

}

public void run(){

int i=0;

while(i<3){

q.put(i++);

}

}

Class consumer implements Runnable

Q q;

Consumer(Q q){

this.q=q;

new Thread(this, "consumer").start();

}

public void run(){

int i=0;

while(i<3){

int r=q.get();

System.out.println("consumed:" + r);

i++;

}

}

✓ WIP

Class PC

public static void main(String args[]){

System.out.println("Aditya Sharma 1Bh22cs021");

Q q=new Q();

new Producer(q);

new Consumer(q);

System.out.println("Press control-c to stop.");

^3