

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



OBJECT ORIENTED JAVA PROGRAMMING

Submitted by:

SHREE SANKET

1BM22CS261

Faculty incharge:

Dr Seema Patil

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

**B.M.S COLLEGE OF ENGINEERING DEPARTMENT OF COMPUTER
SCIENCE AND ENGINEERING**

Shreya Pancreti Kulkarni
IBM22CS261

Object Oriented Java

Feb-27 : Lab Internals

Quiz : 12 Feb

3 → observation

2 - Viva

8 → output

2 → Pdf upload

I) write a program in java to find area of a rectangle using parameter

→ 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.println("length of rectangle = "  
    + length);
```

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

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

}

}.

Lab 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, displays a message stating that there are no real solutions.

→ import java.util.Scanner;

class Quadratic

{

int a, b, c;

double r1, r2, d;

void getd()

{

Scanner s = new Scanner(System.in);

System.out.println("Enter the coefficient");

of a, b, c");

a = s.nextInt();

b = s.nextInt();

c = s.nextInt();

};

void compute()

{

while (a == 0)

System.out.println("Not a quadratic equation");

System.out.println("Enter a non zero value for a:");

Scanner s = new Scanner

(System.in);

a = s.nextInt();

$$d = b^2 - 4ac$$

{ if ($d == 0$)

$$r1 = (-b) / (2a)$$

System.out.println("Roots are real & equal");

System.out.println("Root1 = Root2 = " + r1);

}

else if ($d > 0$)

{

$$r1 = ((-b) + (\text{Math.sqrt}(d))) / (2a);$$

$$r2 = ((-b) - (\text{Math.sqrt}(d))) / (2a);$$

System.out.println("Roots are real & distinct");

System.out.println("Root1 = " + r1 + "
Root2 = " + r2);

}

else if ($d < 0$)

{

System.out.println("Roots are imaginary");

$$r1 = (-b) / (2a);$$

$$r2 = \text{Math.sqrt}(-d) / (2a);$$

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

System.out.println("Root2 = " + r1 + " - i" + r2);

}

}

}

class QuadraticMain

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

```
{  
    Quadratic q = new Quadratic();  
    q.getd();  
    q.compute();  
}
```

?

out: Enter the coefficients of a,b,c

4

6

8

Roots are Imaginary

11.5

Root11 = 0.0 + i 0.0

Root1 = 0.0 - i 0.0

USN: IBM 22Q261

Name: Shree Sanket

Rev 12/12/23

Lab - 2

2) Develop a java program to create a class student with members UIN, name, an array credits, an array marks. Include methods to accept and display details and a methods to calculate CGPA of the student.

formula for CGPA

$$CGPA = \frac{\sum [(course\ credits)(grade\ points)]}{\sum [course\ credits]}$$

$$CGPA = \frac{\sum [(course\ credits)(grade\ points)]}{\sum [course\ credits]}$$

```
→ import java.util.*;  
class Subject {  
    int SubjectMarks;  
    int credits;  
    int grade;  
}  
  
class student {  
    student student[7];  
    String name;  
    String USN;  
    double CGPA;  
    Scanner sc = new Scanner(System.in);  
    Student() {  
        int i;  
        Subject = new Subject[8];  
        for (i=0; i<8; i++) {  
            Subject[i] = new Subject();  
        }  
    }  
}
```

```
Public void getstudentDetails() {  
    System.out.println("Enter your name");  
    name = sc.nextLine();  
    System.out.println("USN");  
    USN = sc.nextLine();  
}
```

```
Public void getMarks() {  
    for (int i=0; i<8; i++) {  
        System.out.println("Enter the marks for " +  
                           (i+1));  
        Subject[i].SubjectMarks = sc.nextInt();  
    }  
}
```

```
System.out.println("Enter the credits for "
```

+ (i+1) * "subject");
Subject[i].credits = ScoreEntered(i);
Subject[i].grade = (Subject[i].marks / 10) + 1;
if (Subject[i].grade == 11) {
} else Subject[i].grade = 10;

{
if (Subject[i].grade < 4) {
} else Subject[i].grade = 4;

}

Public void ComputeSGPA() {
int effectiveScore = 0;
int total = 0;
float SGPA = 0;
for (int i=0; i<8; i++) {
effectiveScore += (Subject[i].credits *
Subject[i].grade);
total += Subject[i].credits;
}

System.out.println("Effective Score: " + effectiveScore);

System.out.println("Total: " + total);

SGPA = effectiveScore / total;

System.out.println("SGPA: " + SGPA);

}

Class JavaMain {
public static void main(String args[]) {
}}

Student s = new Student();

```
s.getStudentDetails();
s.getMarks();
s.computeSGPA();
System.out.println("name = " + s.name);
System.out.println("USN = " + s.USN);
```

?

Output :

Enter your name

SHREE SANKET

USN

81BM22C126

Enter the marks for 1 subject

80

Enter the credits of 1 subject

4

Enter the marks for 2 subject

70

Enter the credits of 2 subject

4

Enter the marks for 3 subject

67

Enter the credits of 3 subject

3

Enter the marks for 4 subject

58

Enter the credit of 4 subject

3

Enter the marks for 5 subject

69

Enter the credits of 5 subject

4

Enter the marks for 6 subject

87

Enter the credit of 6 subject

3

Enter the marks for 7 subject

~~Enter~~

59

Enter the credits of 7 subject

1

Enter the marks for 8 subject

50

Enter the credits of 8 subject

1

effective Score : 165

Total Credits : 22

SGPA : 7.0

Name : SOHREE SANJEEV

USN : ~~1~~ BM22CS261

26/12/23

Lab - 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 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 Book {

String name;

String author;

int price;

int numPages;

Scanner s1 = new Scanner(System.in);

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 name, author, price, numPages;

name = "name" + this.name + "\n";

author = "author" + this.author + "\n";

price = "price" + this.price + "\n";

numPages = "no of pages" + this.numPages + "\n";

```
return name + author + price + numPages;
```

```
public class Main {
```

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

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

```
        String name, author;
```

```
        int price, numPages;
```

```
        int n;
```

```
        System.out.println("Enter the number of books: \n");
```

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

```
        Book b[];
```

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

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

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

```
            name = sc.next();
```

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

```
            author = sc.next();
```

```
            System.out.println("Enter the price of book: \n");
```

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

```
            System.out.println("Enter the no. of pages for book: \n");
```

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

```
            b[i] = new Book(name, author, price, numPages);
```

```
            System.out.println("Name: " + b[i].name + " Author: " + b[i].author + " Price: " + b[i].price + " ISBN: " + b[i].ISBN);
```

```
        }
```

```
for (int i=0; i<n; i++) {  
    System.out.println(b[i].toString());
```

{

Output: Enter the number of books: 2

name: SHREE JANAKI ISBN: 1BM22C1201

Enter the name of book:

letus C

Enter the name of author:

ritchy

Enter the price of book:

2000

Enter the no. of pages for book:

690

Enter the name of book:

webdev

Enter the name of author:

Stauny

Enter the price of book

3500

Enter the no. of pages for book:

940

name : letus C

Author-name : ritchy

price : 2000

no of pages : 690

name : webdev
author-name : Stanny
price : 3500
no. of pages : 940

26/12/23

4) Develop a java program to create an abstract class named Shape that contains 2 integers & an empty method named printArea().
Provide 3 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.Scanner;

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

Abstract class Shape extends InputScanner{
 double a;
 double b;
 abstract void getDetails();
 abstract void printArea();
}

class Rectangle extends Shape{
 void getDetails(){

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

a = s.nextDouble();

b = s.nextDouble();

}

```
void printArea() {  
    System.out.print("Area of Rectangle = ");  
    System.out.print(a * b);  
}  
  
class Triangle extends Shape {  
    void getDetails() {  
        System.out.print("Enter a and b");  
        a = s.nextDouble();  
        b = s.nextDouble();  
    }  
}
```

```
void printArea() {  
    System.out.print("Area of Triangle = ");  
    System.out.print((a * b) / 2);  
}
```

```
class Circle extends Shape {  
    void getDetails() {  
        System.out.print("Enter radius");  
        a = s.nextDouble();  
    }  
}  
  
void printArea() {  
    System.out.print("Area of Circle = ");  
    System.out.print(3.14 * a * a);  
}
```

```
class AbstractMain {
    public static void main(String[] args) {
        System.out.println("SHREE SANICET USN: IBM22C(26)");
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle C = new Circle();
        r.getDetails();
        r.printArea();
        t.getDetails();
        t.printArea();
        C.getDetails();
        C.printArea();
    }
}
```

3.

Output: SHREE SANICET USN: IBM22C(26)

Enter length and breadth

20

30

Area of Rectangle = 600.0

Enter a and b

Enter a and b

20

20

Area of Triangle = 200.0

Enter radius

15

Area of Circle = 706.5

X
22/01/24

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 ~~their~~ derive the classes Curr-Acc and Sav-Acc 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 & update the balance.

cheque for min balance. Impose penalty if necessary update the balance.

```
import java.util.*;
```

```
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");
```

```
}
```

```
void display() {
```

```
    System.out.println("Name: " + name + "
```

```
    "In" + "Account NO: " + accno + "\n"
```

```
    "Type: " + type + "\n" +
```

"Balance : " + balance + "\n");

class SavingsAccount extends Account {
 private static int rate = 5;

SavingsAccount(String name, int accno, String type, double balance) {
 Super(name, accno, type, balance);

}

void balanceWithInterest() {

balance += balance * rate / 100;

System.out.println("Balances : " + balan
ce);

}

class CurrentAccount extends Account {

private double minbal = 500;

private double servicecharge = 50;

CurrentAccount(String name, int accno,

String type, double balance) {

Super(name, accno, type, balance);

}

void checkmin() {

if (balance < minbal) {

System.out.println("Balance is
less than min balance, Service charge is

imposed : " + servicecharge);

balance = serviceCharges;
System.out.println("Balance is : "+
balance);

}

Public class Bank {

{ public static void main(String args[])

Scanner s = new Scanner(System.in);
System.out.println("Enter your name:");
String name = s.nextLine();

System.out.println("Enter the account type
(savings or deposit)");

String type = s.next();

System.out.println("Enter the account
number:");

int accno = s.nextInt();

System.out.println("Enter the initial
balance:");

double balance = s.nextDouble();

Double amount1, amount2;

Account Acc = new Account(name, accno
+ type, balance);

Savings Account sa = new SavingsAccount
(name, accno, type, balance);

Current Account ca = new CurrentAccount
(name, accno, type, balance);

```
while (true) {
    if (acc.type.equals("Savings")) {
        System.out.println("1. Balance -- (1)");
        System.out.println("2. Deposit in account");
        case 1: Compute interest (1);
        Display Detail in 3. Exit (4);
    }
}
```

```
System.out.print("Enter your choice:");
int choice = s.nextInt();
switch (choice) {

```

Case 1:

```
System.out.print("Enter deposit amount");
amount 1 = s.nextDouble();

```

```
sa.deposit(amount 1);
break;
```

Case 2:

```
System.out.print("Enter withdrawal amount");
amount 2 = s.nextDouble();

```

```
sa.withdraw(amount 2);
break;
```

Case 3:

```
sa.balanceWithInterest();
break;
```

Case 4:

```
System.out.print("Details:");
sa.detailsDisplay();
break;
```

Case 5:

```
'((("return";)Nump3_sqpt.00);) i  
Default s;"')  
System.out.println("Invalid choice  
of withdrawl amount.  
(37982.0) listA.pop();  
3
```

Else

```
System.out.println("In - mance - In");  
System.out.println("1. Deposit in 2.  
withdraw in 3. display In 4. exit 0");
```

```
System.out.println("Enter your choice");  
int choice = System.out.nextInt();  
switch (choice) {
```

Case 1:

```
System.out.println("Enter the amount");  
amount1 = sc.nextDouble();  
cu.deposit(amount1);  
break;
```

Case 2:

```
System.out.println("Enter the withdrawal  
amount");  
amount2 = sc.nextDouble();  
cu.withdrawal(amount2);  
break;
```

Case 3:

```
System.out.println("Detail: ");  
cu.display();  
break;
```

Case 4:

return;

default:

System.out.println("x'Devalid choice");

3

Output: Enter your name:

Sanketh

Enter the acc type.

savings

Enter the acc number:

12

Enter the initial balance:

500

- Menu -

1. Deposit

2. Withdraw

3. Compute Interest

4. Display detail

5. Exit

Enter your choice

1

Enter the deposit amount

200

— — — Menu — — —

1. Deposit
2. Withdraw
3. Compute Interest for S
4. Display Details
5. Exit

Enter your choice

2

Enter the withdrawal amount.

100

— — — Menu — — —

1. Deposit
2. Withdraw
3. Compute Interest
4. Display Details
5. Exit

Enter your choice.

3

Balance = 630.0

— — — Menu — — —

1. Deposit
2. Withdraw
3. Compute Interest
4. Display Details
5. Exit

Enter your choice

4

Details :
Name : Sanket
Account No. : 1234567890
Type : Saving
Balance : ₹ 630.0

Withdrawing money from account

Menu

1. Deposit
2. Withdraw
3. Compute Interest
4. Display Details
5. Exit

Enter your choice

5

(9) Write a java program to create an abstract class Bird with abstract methods fly() & makeSound(). Create subclasses Eagle & Hawk that extend the Bird class & implement the respective methods to describe how each bird flies & makes a sound.

→ abstract class Bird {

 abstract void fly();

 abstract void makeSound();

}

class Eagle extends Bird {

 void fly() {

 System.out.println("Eagle soars high
 in the sky.");

}

 void makeSound() {

 System.out.println("Eagle makes
 screeching sound.");

}

}

Class Hawk extends Bird {

 void fly() {

 System.out.println("Hawk glides
 in the air.");

}

void mouseSound();

System.out.println("Hawk glides in
the air.");

}

3

~~3~~

class Bird extends Animal

{ public static void main(String[] args)

{ Eagle e = new Eagle();

e.fly();

e.mouseSound();

Hawk h = new Hawk();

h.fly();

h.mouseSound();

out.println("Shree Shreem 1BM22C1261");

}

Out: Eagle soars high in the sky.

Eagle makes screeching sound.

Hawk glides in the air

Hawk emits a distinct cry.

Shree Shreem 1BM22C1261

- 1) Demonstrate various String Constructor with proper java programs.
- 2) Demonstrate String length, String literal, String concat.

→ public class StringMain
 Public static void main (String [] args) {
 // Constructors.
 System.out.println ("Shor Sanket" + " IBM2242");
 String s = new String ();
 String s2 = new String ("Hello Java");
 System.out.println ("s0 = " + s2);

char mychars [] = {'T', 'a', 'v', 'a'};
 String s3 = new String (mychars);
 System.out.println ("s3 = " + s3);

String s4 = new String (mychars, offset: 0,
 count: 2);
 System.out.println ("s4 = " + s4);

byte b [] = {65, 66, 67, 68};
 String s5 = new String (b);
 System.out.println ("s5 = " + s5);

// String length, String literal, String
 Concat.

String name = "Shree Sanketa Kulkarni";
System.out.println("displaying string length
");

System.out.println("length = "+name.length
());
System.

System.out.println("displaying concatenation
of String");

String branch = "Computer Science
Engineering";

String details = name + branch;

System.out.println("concatenated string
= "+details);

3

Shree Sanketa IBM22C026

Output: s2=Hello Java

s3 = Java

s4 = java

s5 = ABCD

displaying string length
length = 22

displaying concatenation of string.

Concatenated string = Shree Sanketa
Kulkarni Computer Science Engineering

16/01/24

19) Public class GenericStack<T> {

 Private Object[T] StackArray;

 Private int top;

 Private static void final int MaxSize = 10;

 Public GenericStack() {

 StackArray = new Object[MaxSize];

 top = -1;

}

 Public void push(T element) {

 if (top < MaxSize - 1) {

 StackArray[t + top] = element;

 System.out.println("Pushed: " + element);

}

 else {

 System.out.println("Stack is full. Cannot
 push more elements");

}

 Public T pop() {

 if (!isEmpty()) {

 @Supernovings ("checked")

 T element = (T) StackArray[top - 1];

 System.out.println("Popped: " + element);

 return element;

}

 else {

 System.out.println("Stack is empty
 Cannot pop element");

return null;

}

public boolean isEmpty(){
 return top == -1;

}

public static void main(String[] args){
 GenericStack<Integer> integerStack = new
 GenericStack<()>();
 integerStack.push(element: 1);
 integerStack.push(2);
 integerStack.push(3);
 integerStack.pop();

GenericStack<Double> doubleStack = new
GenericStack<()>();
doubleStack.push(1.5);
doubleStack.push(2.5);
doubleStack.push(3.5);
doubleStack.pop();

}

Output: pushed : 1
 pushed : 2
 pushed : 3
 popped : 3
 pushed : 1.5
 pushed : 2.5
 pushed : 3.5
 pushed : 3.5.

* Strings

1) Sanketh

Swati

Sresh

2) length = 5

Shree

3) a=444

4) RMSCR

5) 65

66

67

abc.

6) True

false

false

true

7) String is matched

8) true

true

9) false

true

10) true

false

- 11) apple ball can dig int free green hen rice
fug kite lift man her Orange is parrot
queen ring Stan free

12) 11.9 80.40. 53.6 17 87.19.2 10

13) This is a turtle. The is too

14) Hello world

15) Commage

16) Hello friends

17) Student 1
name : Shree Sankita
Reg no. : 1008
SEM = 3
CGPA = 9.3

18) char at 31 is 'x'
abc
reverse inobd

19) Carea = 28.26
Cperi = 18.84
Tarea = 40
Tper = 83

Ji
19101124

Labs - 6

Create a package CIE which has two classes - Student and Internals derived from student has an array that stores the Internals 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. Export the two packages in a file that declares the final marks of n students in all five courses.

→ CIE

Internal.java

Package cie;

import java.util.Scanner;

Public class Internal Extends Student {

Public int InternalsMarks[] = new int[5];

Scanner sc = new Scanner (System.in);

Public void setCIE () {

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

System.out.println ("Enter CIE marks
of Subject ")+(i+1));

internalsMarks[i] = sc.nextInt();

3

?

student.java

```
package cie;
import java.util.Scanner;
public class Student {
    public int sem;
    public String USN, name;
    Scanner sc = new Scanner(System.in);
    public void setStudentDetails() {
        System.out.println("Enter your name:");
        name = sc.nextLine();
        System.out.println("Enter your USN:");
        USN = sc.nextLine();
        System.out.println("Enter your Sem:");
        sem = sc.nextInt();
    }
    public void getDetails() {
        System.out.println("Name: " + name);
        System.out.println("USN: " + USN);
        System.out.println("Sem: " + sem);
    }
}
```

See

External.java

```
package cie;
import cie.Student;
import java.util.Scanner;
```

public class External { Extends Testinternal {
 public int seeMarks [] = new int [5];
 public int final Marks [] = new Test [5];
 Scanner sc = new Scanner (System. in);
 public void setSBE () {
 for (int i = 0; i < 5; i++) {
 System.out.print ("Enter see marks
 of subject " + (i + 1));
 seeMarks [i] = sc.nextInt ();
 }
}

public void ComputeFinal () {
 for (int i = 0; i < 5; i++) {
 final Marks [i] = InternalMarks [i] + see
 Marks [i] / 2;
 }
}

public void displayMarks () {
 for (int i = 0; i < 5; i++) {
 System.out.println ("Subject " + (i + 1) +
 " = " + final Marks [i]);
 }
}

DemoMain

DemoMain.java

Package DemoMain

```
import See.External;
public class DemoMain {
    public static void main (String args) {
        External Obj = new External();
        Obj.setStudentDetails();
        Obj.getDetails();
        Obj.setCIE();
        Obj.setSEE();
        Obj.computeFinal();
        Obj.displayMarks();
        System.out.println("IBM22CS261" + "IBM22CS261");
    }
}
```

Output:

Enter your name: Shree Saiketh

Enter your USN: IBM22CS261

Enter your sem: 3

Name: Shree Saiketh

USN: IBM22CS261

Enter sem: 3

Enter your CIE marks of subject 1
40

Enter CIE marks of subject 2
40

Enter CIE marks of subject 3
38

Enter CIE marks of subject 4
35

Enter see marks of subject 1

36

Enter see marks of subject 1 & 2

99

Enter see marks of subject 2 & 3

98

Enter see marks of subject 3

97

Enter see marks of subject 4

95

Enter see marks of subject 5

95

Subject 1 = 89

Subject 2 = 89

Subject 3 = 86

Subject 4 = 82

Subject 5 = 83

Shree Sankar 1RM22C1201

Lab - 7

* Write a program that demonstrates handling of exception in inheritance. 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" & throws the exception wrongAge() when the input age < 0. In son class, implement a constructor that can both father & son's age & throws an exception if son's age is \geq father's age.

→ import java.util.Scanner;

```
class WrongAge extends Exception  
public WrongAge (String s) {  
    super(s);
```

class Father {

int fatherAge;

```
Scanner sc = new Scanner (System.in);  
public void validAge () throws Exception  
{ System.out.print ("Enter father's  
age:");
```

fatherAge = sc.nextInt();

if (fatherAge == 0) {

throw new wrongage ("Invalid
father age, Age cannot be negative.");

class Son extends Father

int sonAge;

Scanner scanner = new Scanner(System.in);

public void validate() throws

wrongAge

System.out.println("Enter Sons age :");

sonAge = scanner.nextInt();

super.validate();

if (sonAge >= fatherAge) {

throws new wrongAge("Son age

cannot be greater than fathers age")

}

else if (fatherAge == sonAge) {

throws new wrongAge("fathers and sons

age cannot be the same ");

}

else if (sonAge < 0) {

throws new wrongAge("Invalid

Son age");

}

}

}

Public class Mymain

public static void main(String[] args)

{ Son obj = new Son();

try {

obj.validate();

}

catch (wrongAge e) {

System.out.println("exception");

e.getMessage());

System.out.println("Shree Sanketa, IBM22C021");

Output: Enter sons age

20

Enter fathers age: 24

Exception: Sons age & fathers age
cannot be same.

✓ Shree Sanketa. IBM22C021

30.01.24

Lab - 8

Q 8) 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.

```
→ class Bmse Extends Thread {  
    public void run() {  
        for (int i = 1; i <= 10; i++) {  
            System.out.println(i);  
            System.out.println("BMSCE");  
            try {  
                Thread.sleep(10000);  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

```
class Cse Extends Thread {  
    public void run() {  
        for (int i = 1; i <= 10; i++) {  
            System.out.println("CSE");  
            try {  
                Thread.sleep(2000);  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

```
public class MyMain {
    public static void main(String[] args) {
        BMSCE Obj1 = new BMSCE();
        CSE Obj2 = new CSE();
        Obj1.start();
        Obj2.start();
    }
}
```

3

2.

Out: ~~EE~~

BMSCE

CSE

CSE

CSE

CSE

CSE

2

BMSCE

CSE

CSE

CSE

CSE

CSE

3

BMSCE

4

BMSCE

5

~~BMSCE~~

6

~~BMSCE~~

7

2. BMSCE 10

8 BMSCE 10

9 BMSCE 10

10

BMSCE

Demonstrate Timer process communication &
deadlock \rightarrow class of

int n

Synchronized into get() {

System.out.println("got " + n);
return; }

Synchronized void put(int n) {
this.n = n;

System.out.println("put " + n);
}

Class procedure implements Runnable {

Q. p;

procedure (Q. p) {

this.q = q;

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

}

public void run() {

int i = 0;

while (i < 8) {

q = put(i++);

}

Class consumer implements Runnable {

char q;

Consumer (Q. q) {

this.q = q;

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

3.

Public void run () {

int i = 0;

while (i < 6) {

int r = q.get();

i++;

}}}

Class pc {

public static void main (String args[]) {

Qq = new Q();

new "procedure" (q);

new Consumer (q);

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

System.out.println ("Shree Saurabh IBM 22CS261")

Output:

Put 0

Got 0

Put 1

Got 1

Put 2

Got 2

Put 3

Got 3

Put 4

Got 4

Put 5

Got 5

Shree Saurabh IBM 22CS261

Lab-10-B

class A {

 synchronized void foo (B b) {

 String name = Thread.currentThread().getName();

 System.out.println(name + " entered A.foo");
 try {

 Thread.sleep(1000);

}

 } catch (Exception e) {

 System.out.println("A interrupted");

 } System.out.println(name + " trying to call

 B.last());

 b.last();

}

 void last() {

 System.out.println("Inside A.last");

}

class B {

 synchronized void bar (A a) {

 String name = Thread.currentThread().

 getName();

 System.out.println(name + " entered B-
 bar");

 try {

 Thread.sleep(1000);

}

 } catch (Exception e) {

 System.out.println("B interrupted");

System.out.println("name + " trying to call
A.last()")

A.last(): time business logic

void last(){}

System.out.println("inside A.last")

}

public class Deadlock implements Runnable

A a = new A();

B b = new B();

Deadlock()

Thread.currentThread().setName

("Main Thread"); Thread t = new Thread

+ start();

(this, racing in other thread)

a.foo(b);

System.out.println("Back in Main
Thread");

3

public void run(){}

b.bar(a);

System.out.println("Back in Other
Thread");

4.

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

5

?

Output:

Racing in other thread entered Babar.
Main thread entered ~~A. last~~ ~~B. last~~

Racing in other thread trying to call.
~~A. last ()~~

~~Inside A. last~~

Back in the other thread.

Main thread trying to call ~~B. last ()~~

~~Inside A. last~~

Back in Main thread.

~~Fri
13.02.24~~

Lab - 9

Q) Write a program that creates a user interface to perform integer divisions off num1 & num2 is displayed in the Result field when the Divide button is clicked if Num1, Num2, were not an integer, the program should throw an arithmetic exception. If Num2 were zero, the program would throw an arithmetic exception. Display the exception in a message dialog box.

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
  
class SwingDemo {  
    class SwingDemo {  
        SwingDemo () {  
            JFrame frm = new JFrame("Divide APP");  
            frm.setSize(275, 150);  
            frm.setLayout(new FlowLayout());  
            frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
            JLabel lab = new JLabel("Enter the divisor and dividend:");  
  
            JTextField aJtf = new JTextField(8);  
            JTextField bJtf = new JTextField(8);
```

JButton button = new JButton("Calculate");
JLabel Err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel auslab = new JLabel();

for

Jfrm.add(Err);
Jfrm.add(alab);
Jfrm.add(aJtf);
Jfrm.add(bJtf);
Jfrm.add(button);
Jfrm.add(alab);
Jfrm.add(blab);
Jfrm.add(auslab);

ActionListener listener = new ActionListener()
{

public void actionPerformed(ActionEvent
eot) {

System.out.println("Action event from
a text field");

}

};

aJtf.addActionListener(listener);
bJtf.addActionListener(listener);

button.addActionListener(new Actionlistener)
) {

public void actionPerformed(ActionEvent evt) {

try {
int a = Integer.parseInt(tf1.getText());
int b = Integer.parseInt(tf2.getText());

if (b == 0) {

throw new ArithmeticException();

}

int ans = a / b;

alab.setText("lnA = " + a);

blab.setText("lnB = " + b);

anslab.setText("Ans = " + ans);

Err.setText(" ");

}

catch (NumberFormatException e) {

Err.setText("Enter Only Digits!");

alab.setText("");

blab.setText("");

anslab.setText("");

}

Catch (ArithmeticException e) {

Err.setText("B should be NON zero!");

alab.setText("");

blab.setText("");

anslab.setText("");

}

};

```
frm.setVisible(true);
```

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

out: javac swingDemo.java
java swingDemo.
Action Event from a Text field.

Enter the divider & dividend:

calculate

Enter the divider & dividend:

calculate

$$A=10 \quad B=2 \quad Ans=5$$

20.02.24

Functions

TFrame:

Represents the main window of a GUI application. It can provide functionalities to create, manipulate & manage top-level containers. we can add buttons Text field using this.

setSize:

setSize (int width, int height) is a method of the JFrame class used to set size of the frame window in pixels.

setLayout:

setLayout (Layout Manager layout) is a method of the Container which choose how components inside the window are arranged.

Label:

Displays the text or images on the window.

TextField

Provides an editable text box for user input.

Add Frames

Used to add new frame.

20.02.24