INDEX

Name Shreeya Soni

~~Standard~~ Semester 3rd Section E Roll No. 1BM22CS268

Subject Object Oriented Java Programming

SL No.	Date	Title	Page No.	Teacher Sign / Remarks
1.	12.12.23	Lab-1 (Quadratic)	1-2	
2.	19.12.23	Lab-2 (SGPA)	3-6	
3.	26.12.23	Lab-3 (Books)	7-9	
4.	2.01.24	Lab-4 (Abstract Class)	10-12	
5.	9.01.24 16.01.24.	Lab-5 (Bank Account) (Saving / Current) String Generics	13-18	
6.	23.01.24	Lab-6 (Packages)	19-23	
7.	31.01.24	Lab-7 (Exception)	24-26	
8.	6.02.24	Lab-8 (Threads)	27-28	
9.	20.02.24	Lab-9 (Interface to perform Integer division	34-38	
10.	6.02.24	Lab-10 (IPC and Deadlock)	29-33	

Bajna Gold
Date: 12.12.23

LAB

~~Additional~~ Program -

(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 discriminate b^2-4ac is negative, display 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 coefficients 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*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);
  }
}
}
class Quadratic Main
{
  public static void main(String args[])
  {
    Quadratic q = new Quadratic();
    q.getd();
    q.compute();
  }
}
```

Output –

(1) Enter the coefficients of a, b, c
2 3 4
Roots are imaginary
Root1 = 0.0 + i 1.198957
Root1 = 0.0 - i 1.198957

(2) Enter the coefficients of a, b, c
0 0 0
Not a quadratic equation
Enter a non zero value for a
5
Roots are real and equal
Root1 = Root2 = 0.0

for 1, 2, 3

NAME - SHREYA SONI
USN - 1BM22CS268

13/12/23

Lab Program - 2

① 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.

SGPA -

$$SGPA = \frac{\sum [(\text{course credits})(\text{grade Points})]}{\sum [(\text{course Credits})]}$$

Considering all courses registered.

```
import java.util.Scanner;
class Subject
{
int subjectMarks;
int credits;
int grade;
}
class Student
{
Subject subject[];
String name;
String USN;
double SGPA;
Scanner s;
Student()
{
int i;
subject = new Subject[9];
for(i=0; i<9; i++)
```

Bafna Gold
Date: 4 Page:

```
{
  Subject[i] = new Subject();
  s = new Scanner(System.in);
}}
void getStudentDetails()
{
  System.out.println("Enter your name");
  name = s.next();
  System.out.println("Enter your USN:");
  USN = s.next();
}
void getMarks(){
  for(int i=0; i<8; i++)
{
  System.out.print("Enter marks for subject "+(i+1)+":");
  Subject[i].subjectMarks = s.nextInt();
  System.out.print("Enter your credits for subject"+(i+1)+":");
  Subject[i].credits = s.nextInt();
  Subject[i].grade = (Subject[i].subjectMarks/10)+1;
  if (Subject[i].grade==1)
    Subject[i].grade=10;
  if (Subject[i].grade<=4)
    Subject[i].grade=0;
}
}
void computeSGPA()
{
  int effectiveScore=0;
  int totalCredits=0;
  for(int i=0; i<8; i++)
  {
```

```
effectiveScore += (Subject[i].grade * Subject[i].credits);
totalCredits += Subject[i].credits;
}
SGPA = (double) effectiveScore / (double) totalCredits;
}
}
class Main
{
  public static void main(String args[])
{
  Student s1 = new Student();
  s1.getStudentDetails();
  s1.GetMarks();
  s1.ComputeSGPA();
  System.out.println("Name: " + s1.name);
  System.out.println("USN: " + s1.USN);
  System.out.println("SGPA: " + s1.SGPA);
  }
}
```

Output -

```
Enter your name: SUREYA
Enter your USN: 1BM22CS268
Enter marks for subject 1: 98
Enter your credits for subject 1: 4
Enter marks for subject 2: 90
Enter your credits for subject 2: 2
Enter marks for subject 3: 97
Enter your credits for subject 3: 2
Enter marks for subject 4: 98
Enter your credits for subject 4: 3
```

Enter marks for subject 5 : 90
Enter your credits for subject 5 : 3
Enter marks for subject 6 : 92
Enter your credits for subject 6 : 4
Enter marks for subject 7 : 81
Enter your credits for subject 7 : 3
Enter marks for subject 8 : 80
Enter your credits for subject 8 : 1

Name- SHREYA
USN: 1BM22CS268
SGPA: 9.2.

(10) Fri 19/12/23

26/12/23

Lab Program - 3

Q. 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;
    public 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="Book name:" + this.name + "\n";
        String author="Author name:" + this.author + "\n";
        String price="Price:" + this.price + "\n";
        String numpages="Number of pages:" + this.numpages +
                "\n";
        return name+author,price+numPages;
    }
}
```

Bafna Gold
Date: 8 Page:

```
public class Main{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        int n = s.nextInt();

        Book b[] = new Book[n];

        for(int i=0; i<n; i++){
            System.out.print("Enter name of the book: ");
            String name = s.next();
            System.out.print("Enter author of the book: ");
            String author = s.next();
            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();
            b[i] = new Book(name, author, price, numpages);
        }
        System.out.println("\nBook Details:");
        for(int i=0; i<n; i++){
            System.out.println("Book" +(i+1) +":\n" +b[i]);
        }
    }
}
```

Output -

Enter the number of books: 2
Enter name of the book: Java
Enter author of the book: John
Enter price of the book: 250
Enter the number of pages of the book: 100
Enter name of the book: DBMS
Enter author of the book: George
Enter the price of the book: 300
Enter the number of pages of the book: 150

Book details:
Book 1:
Book name: Java
Author name: John
Price: 250
Number of page: 100

Book 2:
Book name: DBMS
Author name: George
Price: 300
Number of pages: 150

SHREYA SONI
1BM22CS268

26/12/23

10

02/01/23

Bafna Gold
Date: 10 Page:

Lab Program - 4

Q. 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 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 getInput();
    abstract void displayArea();
}
class Rectangle extends Shape {
    void getInput() {
        System.out.println("Enter length and width of
            the rectangle: ");
        a = s.nextDouble();
        b = s.nextDouble();
    }
```

```
    void displayArea(){
        System.out.println("Area of Rectangle:" + (a*b));
    }
}
class Triangle extends Shape{
    void getInput(){
        System.out.print("Enter base and height of the
                triangle:");
        a = s.nextDouble();
        b = s.nextDouble();
    }
    void displayArea(){
        System.out.println("Area of Triangle:" + (0.5*a*b));
    }
}
class Circle extends Shape{
    void getInput(){
        System.out.print("Enter radius of the circle:");
        a = s.nextDouble();
    }
    void displayArea(){
        System.out.println("Area of Circle:" +
            (Math.PI*a*a));
    }
}
public class main{
    public static void main(String[]args){
        Rectangle rectangle = new Rectangle();
        Triangle triangle = new Triangle();
        Circle circle = new Circle();
```

Bafna Gold
Date: 12 Page:

```
rectangle.getInput();
rectangle.displayArea();
triangle.getInput();
triangle.displayArea();
circle.getInput();
circle.displayArea();
    }
}
```

Output –

```
Enter length and width of the rectangle: 5 8
Area of Rectangle: 40.0
Enter base and height of the triangle: 6 4
Area of triangle: 12
Enter radius of the circle: 8
Area of circle: 201.0619
```

Name – Shreya Soni
USN – 1BM22CS268

02/01/24

9/01/24

Lab Program - 5

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

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

a) Accept deposit from customer and update the balance.
b) Display the balance.
c) Compute and deposit interest
d) Permit withdrawal and update the balance.
Check for the minimum balance, impose penalty if necessary and update the balance.

⇒

```
import java.util.Scanner;
class Account {
    String customerName;
    String accountNumber;
    String accountType;
    double balance;
Account (String customerName, String accountNumber,
    String accountType, double balance)
    {
```

```
this.customerName = customerName;
this.accountNumber = accountNumber;
this.accountType = accountType;
sc = new Scanner(System.in);
}
Void deposit (){
System.out.println("Enter the deposit amount:");
int dipo = sc.nextInt();
balance += dipo;
}
void withdrawal (){
System.out.println("Enter the withdrawal amount:");
int with = sc.nextInt();
if (with > balance){
System.out.println("Insufficient Balance");
}
else{
balance = with; }
}
void display (){
System.out.println("Customer name:" + customerName);
System.out.println("Account number:" + accountNumber);
System.out.println("Balance:" + balance);
}
void applyinterest (){ }
}
package java_lab;
public class Cur_acct extends Account{
Cur_acct { String customerName, int accountNumber,
String accountType)
{Super(customerName, accountNumber, accountType); }
```

```
void withdrawal(){
  System.out.println("Enter the withdrawal amount:");
  int with = sc.nextInt();
  if (balance <= 2000){
     double pen = balance/(0.06);
  System.out.println("Insufficient balance penalty to
     be paid:" +pen);
  balance += pen;
  }
  else{
       balance = with;
    }
  }
}
```

```
package java_lab;
public class Sav_acct extends Account{
   Sav_acct(String CustomerName, AccountNumber,
          accountType)
     {
Super(CustomerName, AccountNumber, accountType);
  }
void applyinterest(){
  System.out.println("Enter the interest rate:");
  int rate = sc.nextInt();
  double interest = balance*(rate/100);
  balance += interest;
  System.out.println("Balance after interest:"+balance);
  }
}
```

```
package java_lab;
```

Bafna Gold
Date: 16 Page:

```
import java.util.Scanner;
public class Bank{
public static void main(String args[]){
  Scanner sc= new Scanner(System.in);
System.out.println("Enter customer name:");
String customerName = sc.nextLine();
System.out.println("Enter account number:");
int accountNumber = sc.nextLine();
Account Ca = new
Cur_acct (customerName, account number, "Current account");
Account Sa = new
Sav_acct (customerName, accountNumber, "Saving account");
int choice;
while(true){
System.out.println("---- Menu ----");
System.out.println("1. Deposit \n2. Withdrawal \n
   3. Compute interest for saving account \n 4. Display
   account details \n5. Exit");
choice = sc.nextInt();
Switch(choice){
Case 1;
  System.out.println("Enter the type of account:
     \n1. SavingAccount \n2. Current Account");
int acc = sc.nextInt();
if(acc==1){
  Sa.diposit();
  }
else{
    Ca.deposit();
}
  break;
```

```
case 2:
    System.out.println("Enter the type of account:\n
    1. Saving Account \n 2. Current Account");
    int acc1 = sc.nextInt();
    if (acc1==1){
    sa.withdrawal();
    }
    else{
    ca.withdrawal();
    }
    break;
case-3:
    sa.applyinterest();
    break;
case-4:
    System.out.println("Enter the type of account:\n
    1. Saving account \n 2. Current Account");
    int acc2 = sc.nextInt();
    if (acc2==1){
      sa.display();
    }
    else{
    ca.display();
    }
    break;
case 5: Break;
default: System.out.println("Invalid Choice");
        break;}
if (choice==5){
        break;}}
}}
```

Bafna Gold
Date: 18 Page:

Output:

Enter Customer name:
Shreya
Enter account number-
22
--Menu--
1- Deposit
2- Withdrawl
3. Compute interest for saving account.
4. Display account details
5. Exit
1
Enter the type of account:
1. Saving Account 2. Current Account.
1.
Enter deposite amount: 5000
2
Enter the type of account:
1. Saving account 2. Current Account 1.
Enter the withdrawl amount: 2000
4.
Enter the type of account: 1.
Customer name: Shreya
Account Number: 22
Type of account: Saving Account
Balance: 3000
2.
Enter the withdrawl amount: 6000
Insufficient balance. Penalty to be paid: 833.334

Shreya Soni
1BM22CS268

Generics -

Write a Java program to create a generic class Stack which hold 5 integers and 5 doubles values.

Input -

```
import java.util.EmptyStackException;
public class stack <T> {
    private int maxSize;
    private int top;
    private Object [] stackArray;

    public Stack (int size){
    maxSize = size;
    stackArray = new Object [maxSize];
     top = -1;
    }
public void push (T value){
    if (top < maxSize -1) {
      top ++;
      stackArray [top] = value;
    }
    else {
      throw new RuntimeException ("Stack is full");
    }
}
@ Suppress warnings ("Unchecked")
public T pop(){
    if (!isEmpty ()) {
      return (T) stackArray [top--];
} else {
      throw new EmptyStackException();
}}
```

```
public boolean isEmpty(){
    return (top == -1);
}
public int size(){
    return top+1;
}
public static void main(String args[]){
    Stack<Integer> intStack = new Stack<>(5);
    Stack<Double> doubleStack = new Stack<>(5);
    for(int i=0; i<5; i++){
        intStack.push(i);
        doubleStack.push((double)i);
    }
    System.out.println("Integer Stack:");
    for(int i=0; i<5; i++){
        System.out.println(intStack.pop());
    }
    System.out.println("Double Stack:");
    for(int i=0; i<5; i++){
        System.out.println(doubleStack.pop());
    }
}
```

Output -

Integer Stack:	Double Stack:
4	4.0
3	3.0
2	2.0
1	1.0
0	0.0

16/01/24

Shreya Soni
1BM22CS268

23/01/24

Bafna Gold
Date: 19 Page:

Lab Program - 6

Q. 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 derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

=> Input:-

In CIE package, Student.java →

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

In CIE package, Internals.java →

```
package CIE;
import java.util.Scanner;
```

```
public class Internals extends Student
{
  Scanner in = new Scanner(System.in);
  public int [] CIE = new int[5];
  public Internals(String usn, String name, int sem)
  {
    super(usn, name, sem);
  }
  public void get_data()
  {
    for (int i=0; i<5; i++)
    {
      System.out.print("Enter CIE marks of the student
        in subject "+(i+1)+" (out of 50): ");
        CIE[i] = in.nextInt();
    }
  }
}
```

In SEE package, External.java →

```
package SEE;
import java.util.Scanner;
public class External extends CIE.Student
{
  public External(String usn, String name, int sem)
  {
    super(usn, name, sem);
  }
  Scanner in = new Scanner(System.in);
  public int[] see = new int[5];
  Public void get_data()
  {
```

Bafna Gold
Date: 2 page:

```
        for (int i=0; i<5 ; i++)
        {
        System.out.println("Enter SEE marks of the student in
            Subject " +(i+1)+ " (Out of 100) : ");
            See[i] = in.nextInt();
        }
    }
}
```

In default package, final_marks.java →

```
import CIE.*;
import SEE.*;
import java.util.Scanner;
class final_marks
{
public static void main(String args[])
  {
  Scanner in = new Scanner(System.in);
  System.out.print("Enter the number of students: ");
  int n=in.nextInt();
  CIE.Internals obj[] = new CIE.Internals[5];
  SEE.Externals obj[] = new SEE.External[5];
  int total=0;
  for(int i=0;;i<n;i++)
  {
    System.out.println("\n Enter details of Student "
                +(i+1)+": ");
    System.out.println("USN:");
    String USn = in.next();
    System.out.print("Name:");
    String name=in.next();
```

```
System.out.print("Semester:");
int sem = in.nextInt();
obj[i] = new CIE.Internals(usn, name, sem);
obj[i].get_data();
obj1[i] = new SEE.External(usn, name, sem);
obj1[i].get_data();
}
for(int i=0; i<n; i++)
{
System.out.println("\nTotal marks of Student
+(i+1)+" (Out of 100) in:");
for(int j=0; j<5; j++)
{
total = (obj[i].cie[j] + (obj1[i].see[j]/2);
System.out.println("Subject "+(j+1)+" : "+total);
}
}
}
}
```

Output -

Enter the number of Students: 2

Enter details of Student 1:
USN: 1DM19CS39
Name: Shreya
Semester: 3
Enter CIE marks of Student in Subject 1: 45
Enter CIE marks of Student in Subject 2: 50
Enter CIE marks of Student in Subject 3: 48
Enter CIE marks of Student in Subject 4: 39
Enter CIE marks of Student in Subject 5: 36

Enter SEE marks of the Student Subject 1: 89
Enter SEE marks of the Student Subject 2: 85
Enter SEE marks of the Student Subject 3: 97
Enter SEE marks of the Student Subject 4: 94
Enter SEE marks of the Student Subject 5: 91

Enter details of Student 2:
USN: 1BM ISCS164
Name: Nidhi
Semester: 4.
Enter CIE marks of the Student in Subject 1: 43
Enter CIE marks of the Student in Subject 2: 34
Enter CIE marks of the Student in Subject 3: 45
Enter CIE marks of the Student in Subject 4: 42
Enter CIE marks of the Student in Subject 5: 37
Enter SEE marks of the Student in Subject 1: 78
Enter SEE marks of the Student in Subject 2: 76
Enter SEE marks of the Student in Subject 3: 87
Enter SEE marks of the Student in Subject 4: 98
Enter SEE marks of the Student in Subject 5: 91

Total marks of Student 1 (Out of 100) is:
Subject 1: 86
Subject 2: 72
Subject 3: 42
Subject 4: 88
Subject 5: 62

Total marks of Student 2 (Out of 100)
Subject 1: 42
Subject 2: 88
Subject 3: 92
Subject 4: 87
Subject 5: 83

Shreeya Soni
1BM22CS268

23/01/24

31/01/24

Lab-7

Q. Write a program that demonstrate handling of exception 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 {
    public WrongAge(String message)
    {
        super(message);
    }
}
Class Father
{
    int fatherAge;
    Father() throws WrongAge {
        Scanner S, new Scanner(System.in);
        System.out.println("Enter father's age:");
        fatherAge, S.nextInt();
        if (fatherAge < 0)
        {
            throw new WrongAge("Age cannot be negotiated");
        }
    }
    void display()
    { System.out.println("Father's age is:" fatherAge);
    }
}
```

Bafna Gold
Date: Page:

```
class Son extends father{
int sonAge;
Son() throws WrongAge {
  super();
Scanner s= new Scanner(System.in);
System.out.println("Enter Son's age:");
SonAge = s.nextInt();
if (sonAge > fatherAge)
{
  throw new WrongAge("Son's age cannot be greater
  than father's age");
}
else if(sonAge == fatherAge)
{
throw new WrongAge("Son's age cannot be equal to
  father's age");
}
else if(sonAge <0)
{
  throw new wrongAge("Age cannot be negotiated")
}
}
void display()
{
  super.display();
  System.out.println("Son's age is:"+sonAge);
}
}
public class Main
  {
    public static void main(String[] args)
```

```
{
  try
  {
    Son S, new Son();
     S.display();
  }
  catch (WrongAge e)
  {
    System.out.println(e.getMessage());
  }
}
}
```

Output -

Enter Father's Age
22
Enter Son's Age
22
Sons age cannot be same as father's age.

Enter Father's age
30
Enter Son's age
31

Sons age cannot be greater than father's age

Enter father's age -9
Age can't be negative

Enter father age 34
Enter Son's age 8
Father age is 34
Son's age is 8

Shreya Soni
1BM22CS268

31.01.24

6/02/24

Bafna Gold
Date: 27 Page:

Lab-8

Q. 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 BMSCE implements Runnable {
    public void run() {
        while (true) {
            try {
                S.O.P("BMS College of Engineering");
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class CSE implements Runnable {
    public void run() {
        while (true) {
            try {
                S.O.P("CSE");
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Thread t1 = new Thread(new BMSCE());
        Thread t2 = new Thread(new CSE());
        t1.start();
        t2.start();
    }
}
```

Output-

BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE

Shreya Soni
1BM22CS268

Bafna Gold
Date: Page:

Lab-10

Q@ Demonstrate Inter Process Communication and deadlock

IPC

```
class Q {
 int n;
boolean valueSet = false;
Sychronized int get() {
 while (! valueSet)
try {
S.o.P ("\n Consumer waiting \n");
wait();
}
 Catch (Interrupted Exception e) {
S.O.P ("Interrupted Exception caught");
}
S.O.P ("Got !" +n);
valueSet = false;
S.O.P ("\n Intimate Producer \n");
Notify();
 return n;
}
Synchronized void put(int n) {
while (valueSet)
try {
S.O.P ("\n Producer waiting \n");
wait();
}
 Catch (Interrupted Exception e) {
S.O.P ("Interrupted exception caught");
}
This.n = n;
```

```
valueSet = true;
S.O.P("Put:" +n);
S.O.P("\n Intimate Consumer \n");
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<15){
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<15){
int r=q.get();
S.O.P("Consumed:" +r);
i++;
}}
```

```
class PCFixed {
public static void main(String args[]){
Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop");
}
}
```

Output

```
Put: 1
Got: 1
Put: 2
Got: 2
Put: 3
Got: 3
Put: 4
Got: 4
Put: 5
Got: 5
```

Shreya Soni
1BM22CS268

(b) Deadlock -

```
class A{
Synchronized void foo(B b) {
String name = thread.currentThread().getName();
S.O.P(name + "entered A.foo");
try{
  Thread.sleep(1000);
} Catch (Exception e){
S.O.P("A Interrupted");
S.O.P(Name + "trying to call B.last()");
b.last();
}
Void last(){
S.O.P("Inside A.last");
  }
}
Class B{
Synchronized vair bar(A a){
String name = Thread.Current thread().getName();
 S.O.P(name + "entered B.bar");
try{
thread.sleep(1000);
} Catch (Exception e){
S.O.P("B Interrupted");
}
S.O.P(name + "trying to call A.last()"); };
  a.last();
}
Void last(){
      S.O.P("Inside A.last"); }}
```

```
class Deadlock implements Runnable
{
  A a = new A();
  B b = new B();
  Deadlock(){
  Thread.currentThread().setName("MainThread");
  Thread t = new Thread(this, "Racing thread");
  t.start();
  a.foo(b);
  S.o.P("Back in main thread");
}
  public void run(){
    b.bar(a);
    S.o.P("Back in other thread");
}
  Public Static void main(String arg()){
    new Deadlock();
  }
}
```

Output —

```
MainThread entered A.foo
RacingThread entered B.bar.
MainThread trying to call B.last()
Inside A.last
Back in MainThread
RacingThread trying to call A.last()
Inside A.last
Back in other thread
```

Shreya Soni
1BM22CS268

6.02.12

20/02/24

Lab-9

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

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

class SwingDemo{
  SwingDemo(){
  // create jframe Container
  JFrame jfrm = new JFrame("Divider App");
  jfrm.setSize(275,150);
  jfrm.setLayout(new FlowLayout());
// to terminate on close
  jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
//text label
JLabel jlab = new JLabel("Enter the divider and dividant:");
// add text file for both numbers
 JTextField ajtf = new JTextField(8);
 JTextField bjtf = new JTextField(8);
// calc button
 JButton button = new JButton("Calculate");
```

```
// labels
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();
// add in order
jfrm.add(err); //to display error bois
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
ActionListener l = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
      System.out.println("Action event from a text field");
    }
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent evt){
    try{
      int a = Integer.parseInt(ajtf.getText());
      int b = Integer.parseInt(bjtf.getText());
      int ans = a/b;
      alab.setText("\nA = "+a);
      blab.setText("\nB = "+b);
      anslab.setText("\nAns = "+ans);
    }
```

```
catch (NumberFormatException e){
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("Enter only Integers!");
}
catch (ArithmeticException e){
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("B should be Non zero!");
}
}
});
// display frame
jfrm.setVisible(true);
}
public static void main(String args[]){
// create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable(){
    public void run(){
        new StringDemo();
    }
});
}
}
```

Output—

Enter the divider and dividant: [4] [2] [Calculate]

A = 4 B = 2 Ans = 2

Shreya Soni
1BM22CS268

Function

- JFrame.setDefaultCloseOperation(int Operation):
  It is set to exit the application when the user closes JFrame.

- JFrame.add(Component comp):
  This method adds the specified component to this container. It is used to add the JLabel, JTextFields and JButton to the JFrame.

- Integer.parseInt(String s):
  This method parses the specified string as an integer which is used to convert the text entered in the JTextFields to integer.

- JLabel.setText(String Text):
  This method sets the text of the JLabel. which is used to display the result of the division operation.

- JButton.addActionListener(ActionListener):
  This method adds an ActionListener to the JButton which is notified when the button is pressed.

- JTextField.addActionListener(ActionListener):
  The ActionListener is notified when an action occurs, which is when the user presses Enter.

- JFrame.setSize(int width, int height):
  This method sets the size of the JFrames to the specified width and height.

- JFrame.setVisible(boolean b):
  This method makes the JFrame Visible or invisible.

- JFrame.setTitle(String title):
  This method sets the title of the JFrame to the specified string.

- ActionListener.actionPerformed(ActionEvent evt):
  This method is part of the ActionListener interface. The actionPerformed method is called when an action event occurs.

Tue 20.02.24

LAB: 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a 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 coefficients 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*b-4*a*c;
		if(d==0)
		{
			r1 = (-b)/(2*a);
			System.out.println("Roots are real and equal");
			System.out.println("Roo1 = 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("Roo1 = " + 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);
		}
	}
}
```

```
class QuadraticMain
{
	public static void main(String args[])
	{
		Quadratic q = new Quadratic();
		q.getd();
		q.compute();
	}
}
```

LAB: 2

Develop a Java program to create a class Student with members usn, name, an array credits and an array mark. Include methods to accept and display details and a method to calculate SGPA of a student.
//Develop a java program to create a class Student with members usn,name, an array creadits 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 Student{
   String name;
   String usn;
   double SGPA;
   Subject subject[];
   Scanner s;

   Student(){
      int i;
      subject=new Subject[9];
      for(i=0;i<9;i++){
         subject[i]=new Subject();
         s=new Scanner(System.in);
      }

   }

   void getStudentDetails(){
      System.out.println("Enter your Name:");
      name=s.next();
      System.out.println("Enter your USN:");
      usn=s.next();
   }

   void getMarks(){
      for(int 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();

         subject[i].grade=(subject[i].subjectMarks/10)+1;

         if (subject[i].grade==11){
            subject[i].grade=10;
         }
         if (subject[i].grade<=4){
            subject[i].grade=0;
         }

      }
   }

   void computeSGPA(){
      int effectiveScore=0;
      int totalCreadits=0;
```

```
      for(int i=0;i<9;i++){
         effectiveScore += (subject[i].grade*subject[i].credits);
         totalCreadits += subject[i].credits;
      }

      SGPA=(double)effectiveScore/(double)totalCreadits;
   }
}
class Subject{
         int subjectMarks;
         int credits;
         int grade;

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

                  System.out.println("Name:"+s1.name);
                  System.out.println("USN:"+s1.usn);
                  System.out.println("SGPA :"+s1.SGPA);
         }
}
```

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 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, author;
   int price,page_num;

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

   String toStrings(){
      String name,author,price,page_num;
      name="Book name: "+this.name+"\n";
      author="Author name: "+this.author+"\n";
      price="Price: "+this.price+"\n";
      page_num="Number of pages: "+this.page_num+"\n";

      return (name+author+price+page_num);
   }

   public static void main(String args[]){
      int n;
      String name,author;
      int price,page_num;

      Scanner sc=new Scanner(System.in);

      System.out.println("Enter the no. of books:");
      n=sc.nextInt();

      Book b[]=new Book[n];

      for (int i=0;i<n;i++){
         System.out.println("Enter name of book:");
         sc.nextLine();
         name=sc.nextLine();

         System.out.println("Enter author of a book:");
         author=sc.nextLine();

         System.out.println("Enter the price of book:");
         price=sc.nextInt();

         System.out.println("Enter the no. of pages of book:");
         page_num=sc.nextInt();
```

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

        System.out.println("Book Details:");
        for(int i=0;i<n;i++){
            System.out.println("Book "+(i+1)+" :\n"+b[i].toStrings());
        }
    }
}
```

LAB: 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.Scanner;
public class InputScanner {
	Scanner s;
	InputScanner(){
		s=new Scanner(System.in);
	}
}
abstract public class Shape extends InputScanner{
	double a,b;
	abstract void getInput();
	abstract void displayArea();
}
public class Rectangle extends Shape {
	void getInput() {
		System.out.println("Enter the dimension of rectangle:");
		a=s.nextDouble();
		b=s.nextDouble();
	}

	void displayArea() {
		System.out.println("Area of reactangle:"+(a*b));
	}
}
public class Circle extends Shape{
	void getInput() {
		System.out.println("Enter the dimension of circle:");
		a=s.nextDouble();
	}

	void displayArea() {
		System.out.println("Area of circle:"+(3.14*a*a));
	}
}
public class Triangle extends Shape {
	void getInput() {
		System.out.println("Enter the dimension of triangle:");
		a=s.nextDouble();
		b=s.nextDouble();
	}

	void displayArea() {
		System.out.println("Area of triangle:"+((a*b)/2));
	}
}
public class MainMethod {

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

```
                Rectangle R=new Rectangle();
                R.getInput();
                R.displayArea();

                Triangle T=new Triangle();
                T.getInput();
                T.displayArea();

                Circle C=new Circle();
                C.getInput();
                C.displayArea();
        }

}
```

LAB: 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;
public class Account {
	String customer_name;
	int account_no;
	String type_acc;
	double balance=0;
	Scanner sc;

	Account(String customer_name,int account_no,String type_acc){
		this.customer_name=customer_name;
		this.account_no=account_no;
		this.type_acc=type_acc;
		sc=new Scanner(System.in);
	}

	void diposite() {
		System.out.println("Enter the deposit amount:");
		int dipo=sc.nextInt();
		balance+=dipo;
	}

	void withdrawal() {
		System.out.println("Enter the withdrawl amount:");
		int with=sc.nextInt();
		if(with>balance) {
			System.out.println("Insufficient Balance");
		}
		else{
			balance-=with;
		}
	}

	void display() {
		System.out.println("Customer name:"+customer_name);
		System.out.println("Account number:"+account_no);
		System.out.println("Type of Account:"+type_acc);
		System.out.println("Balance:"+balance);
	}
	void applyinterest() {}

}

public class Cur_acct extends Account {

	Cur_acct(String customer_name,int account_no,String type_acc){
		super(customer_name,account_no,type_acc);}
```

```
	void withdrawal() {
		System.out.println("Enter the withdrawl amount:");
		int with=sc.nextInt();
		if (balance<=2000) {
			double pen=balance/(0.06);
			System.out.println("Insuffiecient balance penalty to be paid:"+pen);
			balance+=pen;
		}
		else{
			balance-=with;
		}
	}
}

public class Sav_acct extends Account{
Sav_acct(String customer_name,int account_no,String type_acc){
super(customer_name,account_no,type_acc);
}

void applyinterest() {
System.out.println("Enter the interest rate:");
int rate=sc.nextInt();
double interest=balance*rate;
balance+=interest;
System.out.println("Balance after interest:"+balance);
}
}

import java.util.Scanner;
public class Bank {
	public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter customer name:");
		String cus_name=sc.nextLine();
		System.out.println("Enter account number:");
		int acc_no=sc.nextInt();

		Account ca=new Cur_acct(cus_name,acc_no,"Current Account");
		Account sa=new Sav_acct(cus_name,acc_no,"Saving Account");

		int choice;
		while(true){
			System.out.println("----MENU----");
			System.out.println("1.Deposite\n2.Withdrawl\n3.Compute intereset for Saving
			account\n4.Display account details\n5.Exit");
			choice=sc.nextInt();

			switch(choice) {
			case 1:
			System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
			int acc=sc.nextInt();
			if(acc==1) {
```

```
                    sa.diposite();
                    }
                    else {
                    ca.diposite();
                    }
                    break;
                    case 2:
                    System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
                    int acc1=sc.nextInt();
                    if(acc1==1) {
                    sa.withdrawal();
                    }
                    else {
                    ca.withdrawal();
                    }
                    break;
                    case 3:
                    sa.applyinterest();
                    break;
                    case 4:
                    System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
                    int acc2=sc.nextInt();
                    if(acc2==1) {
                    sa.display();
                    }
                    else {
                    ca.display();
                    }
                    break;
                    case 5:
                    break;
                    default:
                    System.out.println("Invalid Choice");
                    break;
                    }

                    if(choice==5) {
                    break;
                    }
                }
        }
}
```

LAB: 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;
import java.util.Scanner;
public class Student{
   public String usn,name;
   public int sem;
   public void inputStudentDetails(){
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the student usn:");
      usn=sc.nextLine();
      System.out.println("Enter the student name:");
      name=sc.nextLine();
      System.out.println("Enter student semester:");
      sem=sc.nextInt();
   }

   public void disply(){
      System.out.println("Student USN:"+usn);
      System.out.println("Student Name:"+name);
      System.out.println("Student Sem:"+sem);
   }
}

package CIE;
import java.util.Scanner;
public class Internals extends Student{
   public int marks[]=new int[5];
   public void inputCIEmarks(){
      Scanner sc=new Scanner(System.in);
      for(int i=0;i<5;i++){
         System.out.println("Enter the marks for subject "+(i+1)+":");
         marks[i]=sc.nextInt();
      }
   }
}
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends CIE.Internals{
   public int marks[];
   public int finalMarks[];

   public Externals(){
      marks=new int[5];
      finalMarks=new int[5];
   }
```

```
    public void inputSEEMarks(){
        Scanner sc=new Scanner(System.in);
        for(int i=0;i<5;i++){
            System.out.println("Enter subject "+(i+1)+" marks:");
            marks[i]=sc.nextInt();
        }
    }

    public void calculateFinalMarks(){
        for(int i=0;i<5;i++){
            finalMarks[i]=marks[i]/2+super.marks[i];
        }
    }

    public void displayFinalMarks(){
        inputStudentDetails();
        for(int i=0;i<5;i++){
            System.out.println("Subject "+(i+1)+" final marks:"+finalMarks[i]);
        }
    }
}
import SEE.Externals;
class PackageMain{
    public static void main(String args[]){
        int numOfStudent=2;
        Externals finalMarks[]=new Externals[numOfStudent];
        for(int i=0;i<numOfStudent;i++){
            finalMarks[i]=new Externals();
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE Marks:");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE Marks:");
            finalMarks[i].inputSEEMarks();
        }
        System.out.println("Displaying data:\n");
        for(int i=0;i<numOfStudent;i++){
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].disply();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

LAB: 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.

```
public class WrongAge extends Exception{
   WrongAge(String msg){
      super(msg);
   }
}
import java.util.Scanner;
class InputScanner{
   Scanner sc;
   InputScanner(){
      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:"+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(){
      super.display();
      System.out.println("Son's age:"+sonAge);
   }

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

```
        try{
            Son son=new Son();
            son.display();
        }
        catch(WrongAge e){
            System.out.println(e);

        }
    }
}
```

LAB: 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 BMS_College_of_Engineering implements Runnable {
   public void run() {
      while (true) {
         try {
            System.out.println("BMS College of Engineering");
            Thread.sleep(10000); // Sleep for 10000 milliseconds (10 seconds)
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
   }
}

class CSE implements Runnable {
   public void run() {
      while (true) {
         try {
            System.out.println("CSE");
            Thread.sleep(2000); // Sleep for 2000 milliseconds (2 seconds)
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
   }
}

public class ThreadMain {
   public static void main(String[] args) {
      Thread t1 = new Thread(new BMS_College_of_Engineering());
      Thread t2 = new Thread(new CSE());
      t1.start();
      t2.start();
   }
}
```

LAB: 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

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

class SwingDemo{
   SwingDemo(){
      // create jframe container
      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(275, 150);
      jfrm.setLayout(new FlowLayout());
      // to terminate on close
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      // text label
      JLabel jlab = new JLabel("Enter the divider and divident:");

      // add text field for both numbers
      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);

      // calc button
      JButton button = new JButton("Calculate");

      // labels
      JLabel err = new JLabel();
      JLabel alab = new JLabel();
      JLabel blab = new JLabel();
      JLabel anslab = new JLabel();

      // add in order :)
      jfrm.add(err);  // to display error bois
      jfrm.add(jlab);
      jfrm.add(ajtf);
      jfrm.add(bjtf);
      jfrm.add(button);
      jfrm.add(alab);
      jfrm.add(blab);
      jfrm.add(anslab);

      ActionListener l = new ActionListener() {
         public void actionPerformed(ActionEvent evt) {
            System.out.println("Action event from a text field");
         }
      };
      ajtf.addActionListener(l);
      bjtf.addActionListener(l);
```

```
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                try{
                    int a = Integer.parseInt(ajtf.getText());
                    int b = Integer.parseInt(bjtf.getText());
                    int ans = a/b;

                    alab.setText("\nA = " + a);
                    blab.setText("\nB = " + b);
                    anslab.setText("\nAns = "+ ans);
                }
                catch(NumberFormatException e){
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("Enter Only Integers!");
                }
                catch(ArithmeticException e){
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("B should be NON zero!");
                }
            }
        });

        // display frame
        jfrm.setVisible(true);
    }

    public static void main(String args[]){
        // create frame on event dispatching thread
        SwingUtilities.invokeLater(new Runnable(){
            public void run(){
                new SwingDemo();
            }
        });
    }
}
```

LAB: 10

Demonstrate Inter process Communication and deadlock.

```
class Q {
	int n;
	boolean valueSet = false;

	synchronized int get() {
		while(!valueSet)
			try {
				System.out.println("\nConsumer waiting\n");
				wait();
			} catch(InterruptedException e) {
				System.out.println("InterruptedException caught");
			}
		System.out.println("Got: " + n);
		valueSet = false;
		System.out.println("\nIntimate Producer\n");
		notify();
		return n;
	}

	synchronized void put(int n) {
		while(valueSet)
			try {
				System.out.println("\nProducer waiting\n");
				wait();
			} catch(InterruptedException e) {
				System.out.println("InterruptedException caught");
			}
		this.n = n;
		valueSet = true;
		System.out.println("Put: " + n);
		System.out.println("\nIntimate Consumer\n");
		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<5) {
			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<5) {
			int r=q.get();
			System.out.println("consumed:"+r);
			i++;
		}
	}

}

class PCFixed {
	public static void main(String args[]) {
		Q q = new Q();
		new Producer(q);
		new Consumer(q);
		System.out.println("Press Control-C to stop.");
	}
}
```

Deadlock:

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

class Deadlock implements Runnable
{
        A a = new A();
        B b = new B();
        Deadlock() {
            Thread.currentThread().setName("MainThread");
            Thread t = new Thread(this,"RacingThread");
            t.start();
            a.foo(b); // get lock on a in this thread.
            System.out.println("Back in main thread");

        }
        public void run() {
            b.bar(a); // get lock on b in other thread.
            System.out.println("Back in other thread");
        }

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

}
```