Sindhuja Narasimhan 3E
1BM22CS279

INDEX

Lab No	Date	Content	Page No	Remarks
01	12-12-23	Quadratic Roots	1	
02	19-12-23	SGPA Calculator	10	
03	26-12-23	Books	13	
04	02-01-24	Shapes - Abstract class	16	
05	09-01-24	(a) Bank - Account	19	
	16-01-24	(b) Generics	24	
06	23-01-24	Packages	30	
07	30-01-24	Exceptions	35	
08	06-02-24	Threads	38	
09	13-02-24	Deadlock & IPC	40	
10	20-02-24	Interface for Integer Division	44	

⇒ Lab additional programs

a) Biggest of 3 nos
b) factorial
c) palindroms
d) command line argument

12-12-23

12/12/23

1. Demonstrate parseInt
2. Demonstrate scanner class
3. Arrays 1D & 2D
4. factorial of a number

5. Palindrome
6. sum of digits
7. Conversions (widening, narrowing, Promotion
8. Casting incompatible Types

⇒

12/12/23 Lab - Program - 1

1. Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c & use the quadratic formula. If the discriminant b^2-4ac is -ve, display no real soln.

```
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 co-efficients 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 of 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 & equal");
System.out.println("Root1 = Root2 = "+r1);
}
else if (d>0)
{
r1 = ((-b) + (Math.sqrt(d)))/(double)(2*a);
r2 = ((-b) - (Math.sqrt(d)))/(double)(2*a);
System.out.println("Roots are real and distinct");
System.out.println("Root1 = "+r1+" Root2 = "+r2);
}
else if (d<0)
{
System.out.println("Roots are imaginary");
r1 = (-b)/(2*a);
r2 = Math.sqrt(-d)/(2*a);
System.out.println("Root1 = "+r1+" +i"+r2);
System.out.println("Root1 = "+r1+" -i"+r2);
}
}
}
class QuadraticMain
{
public static void main(String args[])
{
Quadratic q = new Quadratic();
```

```
q.getd();
q.computa();
}
}
```

Q to find area of rectangle & verify the same

```
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);
}
}
```

Q Scanner program:

```
import java.util.Scanner;
class Hello World{
public static void main(String args[])
{
int a; float b; String s;
Scanner in = new Scanner(System.in);
System.out.println("Enter a String");
s = in.nextLine();
System.out.println("You entered string "+s);
System.out.println("Enter an integer");
a = in.nextInt();
System.out.println("You entered integer "+a);
System.out.println("Enter a float");
b = in.readFloat();
System.out.println("You entered float "+b);
}}
```

Output 1:

1) Enter the coefficients of a, b, c
2 3 4
Roots are imaginary
Root 1 : 0.0 + i1.198957
Root 2 : 0.0 - i1.198957
Name: Sindhuja Narasimhan
USN: 1BM22CS279

2) Enter the co-efficients of a, b, c
0 5 0 0
~~Roots are quadratic equation~~
Roots are real and equal
Root 1 = Root 2 = 0.0

3) Enter the co-efficients of a, b, c
4 4 4
Roots are imaginary
Root 1 = 0.0 + i0.860625
Root 2 = 0.0 - i0.860075
Name: Sindhuja Narasimhan
USN: 1BM22CS279

Output Rectangle Area:

java Rectangle Area 12 8
length of rectangle = 12
breadth of rectangle = 8
Area of rectangle = 96

Sindhuja Narasimhan
1BM22CS279

19/12/23

Lab 9-12-23

Lab 2: Develop a Java program to create a class Student with members USN, name, an array credits and an array marks. Include methods to accept & display details & a method to calculate SGPA of a student.

$$SGPA = \frac{\sum[(Course\ Credits)(Grade\ Ranks)]}{\sum[(Course\ Credits)]}$$

```
import java.util.Scanner;
class Subject
 {
   int subject Marks;
   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<8; i++){
       subject[i] = new Subject();
       s = new Scanner(System.in);
 }}

void getStudentDetails()
   {
     System.out.print("Enter your Name");
```

```
name = s.next();
System.out.print("Enter your USN: ");
usn = s.next();
}

void getMarks() {
    for(int i=0; i<8; i++)
    {
    System.out.print("Enter marks for subject: ");
    subject[i].subjectMarks = s.nextInt();
    System.out.print("Enter credits for subject: ");
    subject[i].credits = s.nextInt();
    subject[i].grade = (subject[i].subjectMarks/10)+1;
    if (subject[i].grade == 11)
        subject[i].grade = 10;
    else 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 / 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);
  }
}
```

o/p
```
Enter your name : Sindhuja
Enter your USN : 1bm22cs279
Enter marks for subject 1: 90
Enter credits for subject 1: 4
Enter marks for subject 2:
Enter credits for subject 2:
Enter marks for subject 3:
Enter credits for subject 3:
Enter marks for subject 4:
Enter credits for subject 4:
Enter marks for subject 5:
Enter credits for subject 5:
Enter marks for subject 6:
Enter marks for subject 8:
Enter marks for subject 7:
Enter credits for subject 8:
Enter marks for subject 8:
Enter credits for subject 8:
Name: Sindhuja
USN: 1bm22cs279
SGPA: 9.78926
```

1.

LRA 26-12-23

Page No. 13
Date
YOUVA

26/12/2023

Lab - 3

4 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 object books.

```
import java.util.Scanner;
class Book {
    String name;
    String author;
    int price;
    int numPages;

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

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

Output:

Enter the number of book : 2
Enter the name of the book: Java
Enter author of the book : George
Enter the price of the book : 280
Enter the number of pages of the book: 120
Enter name of the book : ComputerOrganization
Enter author of the book : CarlHamcher
Enter the price of the book : 400
Enter the number of pages of the book : 150

Book Details :
Book 1 :
Book name: Java
Author name: George
Price : 280
Number of pages: 120

Book 2:
Book name: ComputerOrganization
Author name: CarlHamcher
Price: 400
Number of pages: 150

Sindhuja Narasimhan
1BM22CS273

UKA 2-1-23

Page No.: 11
Date:

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 class extends the class 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.print("Enter length & 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 the 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();

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

O/p
```
Enter length and width of the rectangle: 2 3
Area of Rectangle 6.0
Enter base and height of the triangle: 6 10
Area of Triangle: 30.0
Enter radius of the circle: 7
Area of the circle: 153.93
```

Sindhya Narasimhan
1BM22CS219

03/08/24

LOD-5

Develop a Java Program to create a class Bank that maintains two kinds of account for its customer, one called saving account...

```
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;
    this.balance = balance;
}
void deposit(double amount)
    {
        balance += amount;
        System.out.println("Deposit successful.
Updated balance: " + balance);
}

void displayBalance()
{
System.out.println("Account Type: " + accountType);
System.out.println("Customer Name: " + customerName);
System.out.println("Account Number: " + accountNumber);
System.out.println("Balance: " + balance);
}
}
```

```
class SavAcct extends Account {
  SavAcct(String customerName, String accountNumber,
  double balance)
  {
  super (customer
```

```
void withdrawal()
{
System.out.println("Enter withdrawal amount");
   int with. sc.nextInt();
   if (with > balance)
    {
     System.out.println("Insufficient balance");
    }
  else {
    balance. with; }
}
```

```
void applyInterest()
  {
  package java_lab;
  public class Cur_acct extends Account {
   Cur_acct (String CustomerName, int accountNumber,
   String account type)
   {
   super (customer Name, Account Number, Account type);
   }

void withdrawal()
  {
  System.out.println("Enter withdrawal amt");
     int with = sc.nextInt()
     if (balance < 2000)
     {

       double pen. balance/(6.06);
System.out.println("Insufficient balance penalty to be paid"
```

```
balance += fen;
else {
    balance = with;
  }
}
}
package java_lab;
public class Sav_account extends Account {
  Sav_acct (String CustomerName, accountNum, accountType)
{
super (Customer Name_ Account Number, account type),
}
void applyInterest()
{
System.out.println("Enter interest rate");
int rate = sc.nextInt();
double interest = balance * (rate/100);
balance += interest;
System.out.println("Balance after interest : " + balance);
}
}

package java_lab;
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 accNum = sc.nextLine();
System.out.println"
  Account ca = new
Cur_acct (customerName, account Num, current account);
```

```
Account so = new
sav_acct [Customer Name, account number, "Savingsaccount"];
int choice;
while (true)
{
System.out.println(" 1. Deposit \n 2. Withdrawal \n
 3 Compute Interest \n 4. Display );
 choice = sc.nextInt();
 switch (choice)
 {
 case 1:
  System.out.println(" Enter the type of account:
       \n 1. Savings \n 2. Current");
  int acc = sc.nextInt();
  if (acc == 1)
   {
   so.deposit();
 }
 else {
    ca.deposit();
}
break;
Case 2:
  System.out.println(" Enter type of account ");
     int acc = sc.nextInt();
    if (acc == 1)
     {
     so.withdrawal();
    }
    else
    {
    ca.withdrawal();
}
break;
```

```
case 3:
    sa.applyInterest();
    break;
case 4:
System.out.println("1.Savings\n2.Current");
    int acc2 = sc.nextInt();
    if(acc2 == 1)
    {
    sa.display();
    }
    else
    ca.display();
    }
    break;
    }
    }
    }
```

9/1/24

Output:

```
Enter Customer Name: Sindhuja
Enter Acc No : 12345
0.-Menue-
1.Deposit
2.Withdrawl
3.Compute interest
4.Display
1
Enter 1.Savings 2.Current
1
Enter amount 5000
Enter your choice
2
Enter 1.Savings 2.Current
1
Enter amount: 2000
Balance amt: 3000
-Menue-
1.Deposit
2.Withdrawl
3.Compute interest
4.Display
4.
1.Savings 2.Current
1.
Customer name: Sindhuja
Acc No : 12345
Balance: 3000
```

Sindhuja Abhishintan
1BM22CS279

16/01/24

Lab 6

1. String Constructors
O/p: ABCDEF

2. Str length
The length of the string is 13

Str literal
Hello World

Str concat
Str1 : Java
Str2 : Programming
Str3 : Java Programming

1. class String
{
public static void main(String args[])
{
char c[] {'J', 'a', 'v', 'a'};
String s1 = new String(c);
String s2 = new String(s1);
System.out.println(s1);
System.out.println(s2);
}
}

O/p
JAVA
JAVA

8. public class Main
{
public static void main(String[] args)
{
String test = "testString";
String pattern = "te";
System.out.println(test.startsWith(pattern));
pattern = "est";
System.out.println(test.startsWith(pattern));
}
}

Op
true
false

19. abstract Write a java program to create an abstract class Bird with abstract methods fly() and makeSound(). Create subclasses Eagle and Hawk

abstract class Bird {
abstract void fly();
abstract void makeSound();

class Eagle extends Bird
{
void fly()
{
System.out.println("Eagle is flying high");
}
void makeSound()
{
System.out.println("Eagle is making a screeching sound");
}
}

```
class Hawk extends Bird {
  void fly()
{
System.out.println("Hawk is flying with a
    speed");
}
void makeSound()
 {
  System.out.println("Hawk is making a crying
      sound");
 }
}

public class Main {
  public static void main(String[] args)
   {
   Eagle eagle = new Eagle();
   Hawk hawk = new Hawk();

   eg eagle.fly();
    eagle.makeSound();
    hawk.fly();
    hawk.makeSound();
  }
}
```

Op.

Eagle is flying high
Eagle is making a screeching sound
Hawk is flying with a speed
Hawk is making a humming sound

Q Write a Java Program to create a generic class Stack which holds 5 integers & 5 double values

```
import java.util.EmptyStack;
public class Stack <T>{
  public int maxsize;
  public int top;
  public 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");
  }
}
 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());
 }
}
}
```

O/p

Int Stack: 4 3 2 1 0

Double Stack: 4.0 3.0 2.0 1.0 0.0

Sindhuja Narayanan
1BM22CS273

23/01/24

Lab - 7

Q Create a package CIE which has two classes. Student & Internals. The class student has members like usn, name and sem.

Student.java

```
package CIE;
public class Student
{
    public String USN;
    public String 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 student name:");
        name = sc.nextLine();
        System.out.println("Enter student semester:");
        sem = sc.nextInt();
    }
public void display()
    {
    System.out.println("Student USN:" + usn);
    System.out.println("Student Name:" + name);
    System.out.println("Student Sem:" + sem);
    }
}
```

Internals.java

```
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();
  }
}
}
```

Externals.java

```
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: ");
 }
}
}
```

Main.java

```
import SEE.Externals;
class Package Main {
 public static void main(String args[])
  {
   int numOfStudent = 2;
   Externals finalMarks[] = new Externals[numOfStudent];
   for(int i=0; i<numOfStudent; i++)
    {
```

```
finalMarks[i] = new FinalMarks();
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].displayFinalMarks();
}
}
}
```

O/p

```
Enter student USN: 2023BM302541
Enter student name: Sindhuja
Enter student semester: 3
Enter CIE marks:
Enter 5 marks: 45 44 46 42 49
Enter SEE marks
98 95 96 88 85

Enter student USN: 1BM22CS279
Enter student name: Shreya
Enter student semester: 3
Enter CIE marks: 50 49 48 43 45
Enter SEE marks: 98 96 95 98 100

Displaying Data
Student Name: Sindhuja
```

Student USN: 2023BMSD2541
Semester 3
Marks 1 : 94
Marks 2 : 91
Marks 3 : 94
Marks 4 : 66
Student Name: Shreya
Student USN: 1BM22CS279
Semester: 3
Marks 1 : 99
Marks 2 : 97
Marks 3 : 95
Marks 4 : 94

30/11/24

Sindhya Narasimhan
1BM22CS279

30/a/24

Lab 8.

Write a program that demonstrates handling exception in inheritance tree. Create a base class called "Father" and derived class called "SON" which extends the base class.

```
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 negative");
    }
}

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

```
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 negative");
    }
}
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());
    }
}
}
```

O/P:

```
Enter Father's age
22
Enter Son's age
22
Son's age cannot be same as father's age

Enter Father's age
30
Enter Son's age
31
Son's 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's age is 34
Son's age is 8.
```

30/11/24

Sindhuja Narasimhan
1BM22CS279

Lab - 9

1. WAP 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 implements Runnable {
    public void run() {
    while(true) {
        try {
            System.out.println("BMS college of Engineering");
        Thread.sleep(10000);
    }
    Catch(InterruptedException e) {
        e.printStackTrace();
    }
    }
}
}

class CSE implements Runnable {
    public void run() {
        while(true) {
        try {
            System.out.println("CSE");
        Thread.sleep(2000);
    }
    catch(Interrupted Expection e) {
        e.printStackTrace();
    }
    }
}
}
```

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

Op

```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
```

2. Demonstrate Inter process communication and deadlock

⇒ Implementation of producer and consumer

```
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("Interrupted Exception 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("Interrupted Exception caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n");
notify();
}
```

```
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();
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");
}
}
```

O/p
```
PUT: 1
GOT: 1
PUT: 2
GOT: 2
PUT: 3
GOT: 3
PUT: 4
GOT: 4
PUT: 5
GOT: 5
```

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");
}
void last() {
 System.out.println("Inside A.last");
}
}
class Deadlock implements Runnable
 {
  A a. new A()
  B b. new B();
```

```
DeadLock(){
 Thread.currentThread().setName("Main");
 Thread t = new Thread(this, "Racing");
  t.start();
}

public void run(){
 b.bar(a);
 System.out.println("Back in other thread");
}
public static void main(String args[]){
  new DeadLock();
}
}
```

Op Main Thread entered A.foo
Racing Thread entered B.bar
Inside A.last
Back in main thread
Racing Thread
Main Thread trying to call B.last()
Inside A.last
Back in mainthread.

Sindhuja Varasimhan
1BM22CS273

20/02/24

Lab - 10

Q Write a program that creates a user interface to perform integer divisions. The user enters two nos. in the ~~next~~ text fields Num1, Num2. The division of Num1 & Num2 is displayed in the Result field when the Divide button is clicked.

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

class SwingDemo {
    SwingDemo() {
    JFrame jfrm = new JFrame("Divides App");
    jfrm.setSize(275, 150);
    jfrm.setLayout(new FlowLayout());
    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
     JLabel jlab = new JLabel("Enter the divider
                              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 anslab = new JLabel();

     jfrm.add(err);
     jfrm.add(jlab);
     jfrm.add(ajtf);
     jfrm.add(bjtf);
```

```
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("\n Ans: "+ans);
  }
  catch(NumberFormatException e) {
      alab.setText("");
      blab.setText("");
      anslab.setText("");
      err.setText("Enter Only Integer!");
  }
  catch(ArithmeticException e) {
      alab.setText("");
      blab.setText("");
      anslab.setText("");
```

```
err.setText("B should be Non zero!");
}
}
});

frm.setVisible(true);
}

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

Op Enter divider and dividend
8 2
Calculate A.8 B.2 Ans. 4

Q Functions used:

1. JFrame.setDefaultCloseOperation(int operation): This method sets the default close operation for the JFrame.

2. JFrame.add : This method adds components to the JFrame. here it is used to add the JLabel, JTextFields & JButton to JFrame.

3. JTextField.addActionListener: This method adds an actionListener to the JTextField. When the user presses Enter in the JTextField, the actionPerformed method the ActionListener is called.

4. JButton.addActionListener: This method adds an actionListener to the JButton.

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

6. JLabel.setText: This method sets the text of JLabel. It is used to display the result of division operation.

7. SwingUtilities.invokeLater(): This method schedules the specified Runnable for execution on the event dispatching thread at a later time.

8. JFrame.setSize: This method sets the size of the JFrame.

9. JFrame.setLayout: This method sets the layout manager for the JFrame. In this case, a FlowLayout is used which arranges components in a left-to-right flow

10 ActionListener.actionPerformed: This method is a part of the action listener interface. It is implemented by ActionListener class, & the action performed is called when an action event occurs.

12/2/24

Sindhya Narasimhan
1BM22CS278

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();
        }

}
```