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

JAVA LAB PROGRAMS

Bachelor of
Engineering in
Computer Science and Engineering

Submitted by:

SHILPA K M

USN Number: 23BECS522

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

INDEX

NAME: Shilpa K.M STD.: CSE SEC.: 3E ROLL NO.: 2023BM302617

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
01	12/12/23	Quadratic Equation	1-3	
02	19/12/23	Calculate SGPA	4-7	
03	26/12/23	N Book objects	8-10	
04	02/01/24	Abstract class	11-13	
05	09/01/24	Inheritence	14-19	
06	23/01/24	Package	28-33	
07	30/01/24	Exception	34-36	
08	6/2/24	Multi Threading	37-38	
09	6/2/24	IPC & Deadlock	45-51	
10	20/2/24	AWT	39-44	

LRA 12-12-23 Date 12/12/23
Page 01

Develop a java programs 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 descriminate $b^2 - 4ac$ 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(&qroot;"Enter the coefficients of
 a, b, c" &(qroots);
  a = s.nextInt();
  b = s.nextInt();
  c = s.nextInt();
 }
  void compute()
  {
  while (a==0)
  {
  System.out.println(&qroot;"Not a quadratic eq?"
  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 & equal");
        System.out.println("Root 1 = Root 2 = " + 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 & distinct");
        System.out.println("Root 1 = " + r1 + "Root 2 = " + r2);
    }
    else if (d<0)
    {
        System.out.println("Roots are imaginary");
        r1 = (-b)/(2*a);
        r2 = Math.sqrt(-d)/(2*a);
        System.out.println("Root 1 = " + r1 + " + i " + r2);
        System.out.println("Root 1 = " + r1 + " - i " + r2);
    }
  }
}

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

o/p

Enter the coefficient of a, b, c
2
3
4
Roots are imaginary
Root 1 = 0.0 + i1.1989578808
Root 2 = 0.0 - i1.1989578

Enter the coefficient of a, b, c
2
4
2
Roots are real & equal
Root 1 = Root 2 = 1

Enter the coefficient of ab, c
2
8
2
Roots are real & distinct
Root 1 = +0.2679
Root 2 = -3.732

Name: Shilpa K M
USN: 2023BMS02617

A 19-12-23

Date 19/12/03
Page 04

3) Develop a java program to create a class student with members usn, name, an array credits & an array marks, include methods to accept & display details & a method to calculate SGPA of students.

SGPA
CGPA

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

considering all courses registered in the semester

$$CGPA = \frac{\sum [(Course\ Credits)(Grade\ points)]}{\sum [(Course\ Credits)]}$$

```
import java.util.Scanner;
class Subject {
    int SubjectMarks;
    int Credits;
    int grade;
}
class Student
{
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Subject[] Subjects;
```

```
Student ()
{
 int i;
 subject = new subject[8];
 for(i=0; i<8; i++){
    subject[i] = new Subject();
 }
  s= new  Scanner (System.in);
}
}
void getStudentDetails()
{
  System.out.print("Enter the Name:");
  name = s.next();
  System.out.print("Enter the USN:");
  usn = s.next();
}
void getMarks()
{
  for(int i=0; i<8; i++){
    System.out.print("Enter the marks for Students:"
                    +(i+1)+":");
    subject[i].subjectMarks = s.nextInt();
    System.out.print("Enter the credits for Students:"
                    +(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 totalCredits = 0;
  for (int i=0; i<8; i++){
    effectiveScore += (subject[i].grade * subject[i].credits);
  }
}

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 : Shilpa K.M
Enter your usn : 1023BMS02617
Enter marks for subject 1 : 98
Enter your credits for subject 1 : 4
(Enter your credits for subject 97)
Enter marks for subject 2 : 97
Enter your credits for subject 2 : 4
Enter marks for subject 3 : 90
Enter your credits for subject 3 : 2
Enter marks for subject 4 : 9
Enter your credits for subject 4 : 3
```

Date __/__/____
Page 07

Enter marks for subject 5 : 9
Enter your credits for subject 5: 3
Enter marks for subject 6 : 81
Enter your credits for subject 6 : 2
Enter your marks for subject 7 : 99
Enter your credits for ~~stude~~ subject 7 : 1
Enter marks for subject 8 : 90
Enter your credits for subject 8 : 1
Name : Shilpa
USN : 2023BM302617

Lab - 03

Date 26/12/23
Page 08

Create a class Book which contains four members: name, author, price, numpages. Include a constructor to set the values for the members. Include methods to set & get the details of objects. Include a toString() method that could display the complete details of book. Develop a java program to create n book objects

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

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

    public static toString(){
        String name, author, price, numPages;
        name = "Book name: " + this.name + "\n";
        author = "Author name: " + this.author + "\n";
        price = "Price name: " + this.price + "\n";
        numPages = "number of pages: " + this.numPages +
        return name + author + price + numPages;
    }
}
```

```
class Book2 {
  Public static void main (String args[]){
   int n ;
   String name ;
   String author ;
   int price ;
   int numPages ;
   System.out.println ("Enter no of Books : ");
   Scanner s = new Scanner (System.in);
  n = s.nextInt();
  Books b[];
   b = new Books[n];
   for (int i=0 ; i<n; i++){
      Scanner p = new Scanner (System.in);
      System.out.println ("Enter Book name: ");
      name = p.next();
      System.out.println ("Enter author: ")
      author = p.next();
      System.out.println ("Enter price: ")
      price = p.nextInt();
       System.out.println ("Enter no of pages : ")
       numPages = p.nextInt();
       b[i] = new Books (name, author, price, numPage
   }
   for (int i=0 ; i<n ; i++){
        System.out.println(b[i].toString());
   }
  }
}
```

Date __/__/__
Page 10

O/P

Enter no. of books 1
Enter Book name dare deed
Enter author name Rao
Enter price 550
Enter no. of pages 230

Book name : dare
author name : Rao
Price : 550
no. of pages : 230

Name : Shilpa k M
USN : 2023BM302617

Lab-04 KA 2-1-24 Date 02/01/24
Page 11

Develop a java program to create an abstract class named shape that contains two integer & an empty method named printArea(). Provide 3 classes named Rectangle, Triangle, circle such that each one of the classes extends the class shape. Each one of classes contain only the method printArea() that prints the area of 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 the dimensions of
        rectangle (length, breadth)");
    a = s.nextDouble();
    b = s.nextDouble();
  }
  void displayArea() {
    System.out.println("Area of Rectangle: "+(a
} }
```

```
class Triangle extends Shape {
    void getInput() {
        System.out.println("Enter dimension of
            Triangle (length, breadth)");
        a = s.nextDouble();
        b = s.nextDouble();
    }
    void displayArea() {
        System.out.println("Area of Triangle: " +
                (a*b*0.5));
    }
}
class Circle extends Shape {
    void getInput() {
        System.out.println("Enter dimension of Circle
                (radius)");
        a = s.nextDouble();
    }
    void displayArea() {
        System.out.println("Area of circle: " + (3.14*a
                *a));
    }
}

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();
    }
}
```

Output

```
Enter the dimension of the rectangle (length & breadth):
2,3
Enter the dimension of the triangle (base & height):
2,4
Enter the dimension of the circle (radius):
3
Area of Rectangle : 6.0
Area of Triangle : 4.0
Area of Circle : 28.259999
```

Name : Shilpa K M
USN : 1DO23BMS02617

Lab - 05

Date 09/01/24
Page 14

* Develop a java program to create a class Bank that maintains two kinds of accounts for its customers, one called savings account & other current account. The savings account provides compound interest & 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 & if balance falls below this level, a service charge is imposed.

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

(a) Accept deposite from customer & update balance.
(b) Display the balance.
(c) Compute & deposite interest
(d) Permit withdrawl & update balance
* Check for the minimum balance, impose penalty if necessary & update the balance.

→
```
import java.util.*;
class account {
    String customer_name;
    int acc_no;
    String type;
    double balance;
```

```
account (String name, int no number,
String type, double balance) {
        customer_name = name;
        acc_no = number;
        this.type = type;
        this.balance = balance;
}
void deposite (double amount) {
    balance += amount;
}
void withdrawl(double amount) {
   if ((balance - amount) > 0) {
        balance -= amount;
   }
   else {
       System.out.println("Insufficient balance");
   }
}
void display() {
    System.out.println("Name:" + customer_name +
"\n" + "account number:" + acc_no + "\n" + "Type:"
+type + "\n" +"Balance:" + balance);
  }
}
class sav_acc extends account {
   private static double rate =5;
   sav_acc (String name, int acc_no, double
balance) {
   super(name, acc_no, "savings", balance);
}
void interest() {
   balance += balance * (rate)/100;
}; System.out.println("balance :" + balance);
```

```
class curracc extends account {
    curr_acc (string name, int acc_no, double
balance) {
    super(name, acc_no, "current", balance);
    }
    private double minbal = 500;
    private double servicecharge = 50;
    void checkmin () {
        if (balance < minbal) {
            System.out.println ("Insufficient balan
-ce so service charges are substracted from
your balance. amount");
            balance -= servicecharge;
            System.out.println ("Balance is :"
+ balance);
        }
    }
}

public class bank {
    public static void main (String args[]) {
        Scanner s = new Scanner ( System.in);
        System.out.println ("Enter the name:");
        String name = s.next();
        System.out.println ("Enter the type (current/
                Savings) :");
        String type = s.next ();
        System.out.println ("Enter the Account
            number:");
        int acc_no = s.nextInt ();
        System.out.println ("Enter the intial
                balance :");
```

```
double balance = s.nextDouble();
int ch;
double amount1, amount2;
account acc = new account(name, acc_no, type, balance);
    savacc sa = new savacc(name, acc_no, balance);
    curracc ca = new curracc(name, acc_no, balance);
while(true){
    if(acc.type.equals("Savings")){
System.out.println("\n Menu\n1. deposit 2. withdraw. 3. Compute interest 4. display");
        System.out.println("Enter the choice:");
        ch = s.nextInt();
        switch(ch){
            case 1: System.out.println("Enter the amount:");
                amount1 = s.nextInt();
                sa.deposite(amount1);
                break;
            case 2: System.out.println("Enter the amount:");
                amount2 = s.nextInt();
                sa.withdraw(amount2);
                break;
            case 3: sa.interest();
                break;
            case 4: sa.display();
                break;
```

```
default : System.out.println ("Invalid input");
          System.exit (0);
        }
    }
    else {
      System.out.println ("\n Menu \n1. deposite 2. with
-draw 3. display");
       System.out.println ("Enter the choice:");
       ch = s.nextInt();
       switch (ch) {
         case 1 :
          System.out.println ("Enter the amount:");
          amount1 = s.nextInt();
          ca.deposite (amount1);
           break;
         Case 2 :
          System.out.println ("Enter the amount:");
          amount2 = s.neatInt();
          ca.withdraw (amount2);
          ca.checkmin();
          break;
        Case 3 : ca.display();
           break;
  default : System.out.println ("Invalid input");
            System.exit(0);
        }
     }
   }
 }
}
```

O/P:-

Enter the name : John
Enter the type(current / savings) : 1
Enter the account number : 1234
Enter the initial balance : 30,000
Menu
1. deposite 2. withdrawal 3. display
Enter the choice : 1
Enter the amount : 80,000
Enter the choice : 2
Enter the amount : 1000
Enter the choice : 3
Name : john
account_number : 1234
Type : current
Balance : 10000

Enter the name : John.B

Name :- John.B
account number : 7272
Type : Savings
Balance : 70000.

9/1/24

Name : Shilpa K M
USN : 2023BMS02617

lab - 6

Date 16/1/24
Page 21

Write a java program using generics Show stack class for 5 integer value & 5 double value

```
import java.util.Stack;
public class stack <E> {
    (E:stck[];
    int top;
    final int size = 5;
    stack() {
     stck = (E[]) new object (size);
     top = -1;
    }
     void push)
    public int maxsize;
    public int top;
    public object [] stack Array;
    public stacks (int size) {
        maxSize = size;
        StackArray = new object(max size);
        top = -1;
    }
public void push (T value) {
    if (top < maxSize - 1) {
        top++;
        stack Array [top] = value;
    }
    else {
        throw new Runtime Exception ("Stack is f
    }
}
```

```
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 <> ();
    Stack <Double> doublestack = new Stack <> ();

for (int i=0 ; i <=5 ; i++) {
    {
    int Stack.push(i)
    doubleStack.push (&double)););
    }
System.out.println("Integer Stack");
    for (int i=0; i <=5 ; i++)
    {
        System.out.println(instack.pop());
}
```

```
System.out.println("Double Stack");
for (int i=0; i<=5; i++){
    System.out.println ~~(Double Stack~~ (double Stack.
        pop());
}
}
```

O/p

Integer Stack

4
3
1
5
2

Double Stack

0.0
9.0
3.0
9.0
1.0

Name : Shilpa K M
USN : 2083BMS02617

④ Using getchars(), extract Bmsce from "welcome to Bmsce college"

```
public class Main {
    public static void main(String[] args) {
        String str = "welcome to Bmsce college";
        char[] chars = new char[6];
        str.getchars(11, 16, chars, 0);
        String bmsce = new String(chars);
        System.out.println(bmsce);
```

o/p → Bmsce.

⑧ Demonstrate startswith() to give output true & false.

```
public class Main {
    public static void main(String[] args) {
        String str = "welcome to Bmsce college";
        boolean startswithBmsce = str.startswith("Bmsce");
        System.out.println(startswithBmsce);

        String str = "Bmsce college welcome you";
        boolean startswithBmsce2 = str2.startswith
                ("Bmsce");
        System.out.println(startswithBmsce2);
    }
}
```

o/p

false
true.

18)

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

19)

```
abstract class Bird {
    abstract void fly();
    abstract void makesound();

class Eagle extends Bird {
    void fly() {
        System.out.println ("Eagle flying in Sky");

    void makssound() {
        System.out.println ("Eagle screeches");
    }
}
class Hawk extends Bird {
    void fly() {
        System.out.println ("Hawk flying silently");
```

```
void makesound() {
    System.out.println("Hawk caws");

public class Main {
    public static void main(String[] args) {
        Bird eagle = new Eagle();
        eagle.fly();
        eagle.makesound();

        Bird hawk = new Hawk();
        hawk.fly();
        hawk.makesound();
}
```

O/P

~~Eagle is flying high~~
Eagle is making a screeching sound
Hawk is flying

op

Eagle flying in sky
Eagle screeches

Hawk flyings silently
Hawk caws.

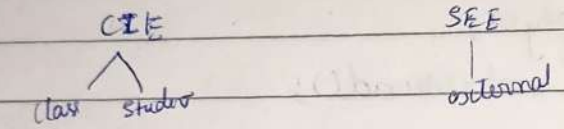
16/11/24

Name: Shilpa KM
USN: 2023BMS02617

Date 23/1/24
Page 28

Lab - 6

Create a package CIE which has two classes Students & 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 current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.



Program :-

```
// Internals.java
package CIE;
import java.util.Scanner;
public class Internals extends Student {
    protected int marks[] = new int[5];

    public Internal() {
    }

    public void inputCIEmarks() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter Internal
                marks for " + name);
        for(int i=0; i<5; i++) {
            System.out.println("Subject " + (i+1) +
                    " marks: ");
            marks[i] = scanner.nextInt();
        }
    }
}
```

```
// Student.java
package CIE;
import java.util.Scanner;
public class Student {
    protected String usn = new String();
    protected String name = new String();
    protected int sem;

    public void inputStudentDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter USN:");
        usn = scanner.next();
        System.out.print("Enter Name:");
        name = scanner.next();
        System.out.print("Enter Semester:");
        sem = scanner.next();
    }
    public void displayStudentDetails() {
        System.out.println("USN:" + usn);
        System.out.println("Name:" + name);
        System.out.println("Semester:" + sem);
    }
}

// Externals.java

package SEE;
import CIE.internals;
import java.util.Scanner;

public class External extends Internals {
```

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

public Externals() {
    marks = new int[5];
    finalMarks = new int[5];
}
public void inputSEEmarks() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter SEE marks for "
                        + name);
    for(int i=0; i<5; i++) {
        System.out.print("Subject " + (i+1) +
                "marks : ");
        marks[i] = scanner.nextInt();
    }
}
public void calculateFinalMarks() {
    for(int i=0; i<5; i++)
        finalMarks[i] = marks[i]/2 + super.
                        marks[i];
}
public void displayFinalMarks() {
    displayStudentDetails();
    for(int i=0; i<5; i++){
        System.out.println("Subject " + (i+1) +
                ": " + finalMarks[i]);
}
}
```

```
// main.java
import SEE.Externals;
public class Main {
    public static void main(String args[]){
        int numofstudents = 2;
        Externals finalMarks[] = new Externals
                                 [numofstudents];

        for(int i=0; i<numofstudents; 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<numofstudents; i++){
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

Output :-

```
Enter USN : bms001
Enter Name : XXXX
Enter Semester : 3
Enter CIE marks :
Enter Internal marks for XXXX
```

Subject 1 marks : 45
Subject 2 marks : 40
Subject 3 marks : 30
Subject 4 marks : 20
Subject 5 marks : 50

Enter SEE marks
Enter SEE marks for XXXX
Subject 1 marks : 44
Subject 2 marks : 39
Subject 3 marks : 47
Subject 4 marks : 38
Subject 5 marks : 50

Enter USN : bms002
Enter Name : YYYY
Enter Semester : 3
Enter CIE marks
Enter Internal Marks for YYYY
Subject 1 marks : 39
Subject 2 marks : 47
Subject 3 marks : 42
Subject 4 marks : 28
Subject 5 marks : 40
Or
Enter SEE marks
Enter SEE marks for YYYY
Subject 1 marks : 44
Subject 2 marks : 50
Subject 3 marks : 48
Subject 4 marks : 38
Subject 5 marks : 35

Displaying data :

USN : bms001
Name : XXXX
Semster : 3
Subject 1 : 67
Subject 2 : 59
Subject 3 : 53
Subject 4 : 39
Subject 5 : 75

USN : bms002
Name : YYYY
Semster : 3
Subject 1 : 61
Subject 2 : 72
Subject 3 : 66
Subject 4 : 47
Subject 5 : 57

Name : Shilpa K M
USN : 2023BMS02617

30/11/24

Lab - 07

Date 30/1/24
Page 34

Write a program the demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" & derived class called "son" which extends the base class. In Father class, implement a constructor which takes age & throws exceptions WrongAge() when the input age < 0. In son class, implement a constructor that cases both father & son's age & throws an exception if son's age is >= father's age

```
import java.util.Scanner;

class WrongAge extends Exception {
    public WrongAge(String e) {
        super(e);
    }
}

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

class Father extends InputScanner {
    int fatherAge;

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

```
public void display () {
    System.out.println("Father's age : " +
                                    fatherAge);
    }
}
class Son extends Father {
    int sonAge;
    public Son() throws WrongAge {
        super();
        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 < 0) {
            throw new WrongAge("Age cannot be
                    negative");
        }
    }
    public void display () {
        super.display();
        System.out.println("Son's age: " + sonAge);
    }
}

public class main {
    public static void main (String[] args) {
        try {
            Son son = new Son();
            son.display();
        }
```

```
        catch (InvalidAge e) {
            System.out.println("Error : " + e.getMessage());
        }
    }
}
```

Output :-

```
Enter father's age : 60
Enter son's age : 22
Father's age : 60
Son's age : 22

Enter father's age : 55
Enter son's age : -18
Error : Age cannot be negative

Enter father's age : 70
Enter son's age : 80
Error : Son's age cannot be greater than father's age
```

Name - Shilpa K M
USN - 2023BM3026I7

30/1/24

Lab - 08

Date 6/2/24
Page 37

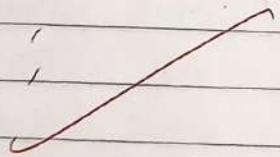
Write a program which creates two threads one thread displaying "BMS college of Engineering" once every ten seconds & another displaying "CSE" once every two seconds.

```
import java.util.concurrent.TimeUnit;

public class Main{
    public static void main (String[] args){
        Thread bms = new Thread (()->{
            while (true){
                System.out.println ("BMS college
                    of Engineering");
                try{
                    TimeUnit.SECONDS.sleep(10);
                }
                catch (InterruptedException e){
                    e.printStackTrace();
                }
            }
        });
        Thread cse = new Thread (()->{
            while (true){
                System.out.println ("CSE");
                try{
                    TimeUnit.SECONDS.sleep(2);
                }
                catch (InterruptedException e){
                    e.printStackTrace();
                }
            }
        });
    }
```

```
Bms.start();
Cse.start();
```

Output :-

```
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
.
.
.
```



Name : shilpa k M
USN : 2023BMS02617

10) Demonstrate Inter Process Communication (IPC) & Deadlock

@ IPC

```
class Q {
int n;
boolean valueSet = false;
Synchronized int get(){
while(!valueSet)
 try {
System.out.println("\n Consumer waiting\n");
wait();
} catch (InterruptedException e){
  System.out.println("InterruptedException caught");
}
  System.out.println("got : " +n);
  valueSet = false;
  System.out.println("\n Intimate Producer\n");
  notify();
  return n;
 }
 synchronized void put(int n){
 while(valueset)
 try{
  System.out.println("\n Producure waiting \n");
  wait();
 } catch (InterruptedException e){
 System.out.println("InterruptedException
                     caught");
 }
 this.n = n;
  valueSet = true;
```

```
System.out.println("Put:" +n);
System.out.println("\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<3) {
q.put(i++);
}
}
}

Class Consumer implements Runnable {
Q q;
Consumer(Q q) {
this.q = q;
new Thread(this, "Consumer").start();
}
public void run() {
    int i=0
while (i<3) {
int x=q.get();
System.out.println("Consumed: " +x);
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: 0
Intimate Consumer
producer waiting
Got: 0
Intimate Producer
Put = 1
Intimate consumer
Producer waiting
consumed: 0
got: 1
Intimate Producer
consumed: 1
Put = 2
Intimate consumer
~~Producer waiting~~
~~Consumed: 1~~
got: 2
Intimate Producer
~~put: 3~~
~~Intimate Consumer~~
~~Producer waiting~~
Consumed: 2

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

```
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);
     thread.
     System.out.println("Back in main thread");
   }

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

Name : Shilpa k M

USN : 2023BMS02617

O/P

Main Thread entered A.foo
RacingThread entered B.bar
Main Thread trying to call B.last()
Inside A.last
Back in main thread
RacingThread trying to call A.last()
Inside A.last
Back in other thread

13.02.24

Name : Shilpa k M
USN : 2023BMS02617

Lab -9

Date 30/2/21
Page 45

10) Write a program that creates a user interface divisions. The user enters two numbers in text fields, Num1 & Num2. The division of Num1 & Num2 is displayed in Result field when the Divide button is clicked if Num1 & Num2 were not an integer, the program would throw an Arithmetic Exception Display exception in a message dialog box.

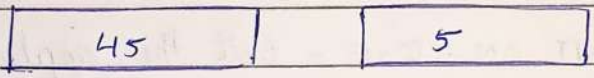
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_
                                      CLOSE);
        JLabel jlab = new JLabel("Enter the divisden
                      & divident:");
        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();
```

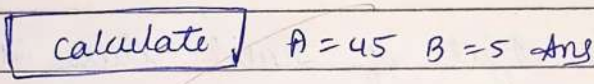
```
jfrm.add(err);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

ActionListener l = new ActionListener(){
public void actionPerformed(ActionEvent ae){
  try{
     int a = Integer.parseInt(ajtf.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.setTest("");
     blab.setTest("");
     anslab.setTest("");
     err.setTest("Enter Only Integers!");
  }
  catch(ArithmaticException e){
     alab.setText("");
     blab.setTest("");
     anslab.setText("");
     err.setText("B Should be Non zero!");
  }
});
```

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

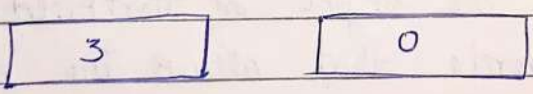
output

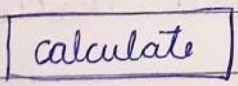
Enter the divisiden & divident:





B should be Non zero!
Enter the divisiden & divident:





Name : Shilpa KM
USN : 2023BM302617

Functions

1) SetSize() :- It is a method is used to set the width & height of a GUI components

2) JFrame() :- The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, textfields are added to create a GUI

3) SetLayout() - method allows you to set the layout of container. The layout manager helps layout the components held by this container.

4) SetDefaultCloseOperation() - method is used to specify one of several options for close button.
JFrame.EXIT_ON_CLOSE - Exit the application

5) JLabel - The object of JLabel class is a components for placing text in a container It is used to display a single line of read only text.

6) JTextField - The object of JTextField class is a text components that allows the editing of single-line text. It inherits JTextComponent class.

7) add(frame) - adds newframe in existing frame

8) SetText() - This method substitutes new text for all or part of text in the text field

This works only with the first line of multi-line text fields.

9) ActionListener - The Java ActionListener is notified whenever you click on the button or menu item. It is notified against ActionEvent. This interface is found in java.awt.event package.

10) SetVisible() - is a method that has return type boolean

20/2/24

LAB 01

```
import java.util.*;
class Roots{
public static void main(String args[]){
double r1, r2;
Scanner sc = new Scanner(System.in);
System.out.println("Enter a,b and c values");
double a = sc.nextDouble();
double b = sc.nextDouble();
double c = sc.nextDouble();
double d  = (b*b)-(4*a*c);
if(d==0){
System.out.println("Roots are real and equal");
r1 = r2 = b/(2*a);
System.out.println("R1="+r1);
System.out.println("R2="+r2);
}
else if (d>0){
Sytem.out.println("Roots are real and distint");
r1 = (-b+Math.sqrt(d))/(2*a);
r2 = (-b-Math.sqrt(d))/(2*a);
System.out.println("R1="+r1);
System.out.println("R2="+r2);
}
else{
Sytem.out.println("Roots are distint and imaginary");
double x = -b/(2*a);
double y = Math.sqrt(-d)/(2*a);
Sytem.out.println("R1="+x+"+i"+y);
Sytem.out.println("R1="+x+"-i"+y);
```

Lab 02

```
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++)
{
subject[i]=new Subject();
s=new Scanner(System.in);
}}
void getStudentDetails()
{
System.out.print("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==11)
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<9;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);
}
}
```

LAB 03

```
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 04

```
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();2
abstract void displayArea();
}

class Rectangle extends Shape{
   void getInput(){
      System.out.println("Enter the dimensions of rectangel(lenght,breadth)");
      a=s.nextDouble();
      b= s.nextDouble();
   }
   void displayArea(){
      System.out.println("Area of Rectangle: "+(a*b));
   }
}

class Triangle extends Shape{
   void getInput(){
      System.out.println("Enter the dimensions of triangle(lenght, breadth)");
      a=s.nextDouble();
```

```
        b=s.nextDouble();
    }
    void displayArea(){
        System.out.println("Area of Triangle: "+(a*b*0.5));
    }
}

class Circle extends Shape{
    void getInput(){
        System.out.println("Enter the dimensions of circle(radius)");
        a=s.nextDouble();
    }
    void displayArea(){
        System.out.println("Area of Circle:" +(3.14*a*a));
    }

}

class Main4 {
    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();
    }
}
```

LAB 05

```
import java.util.*;
class account{
   String customer_name;
   int acc_no;
   String type;
   double balance;
   account(String name,int no,String type,double balance){
      customer_name=name;
      acc_no=no;
      this.type=type;
      this.balance=balance;
   }
   void deposit(double amount){
      balance+=amount;
   }
   void withdraw(double amount){
      if((balance-amount)>0){
         balance-=amount;
      }
      else{
         System.out.println("Insufficient balance");
      }
   }
   void display(){
      System.out.println("Name:"+customer_name+"\n"+"account
number:"+acc_no+"\n"+"Type:"+type+"\n"+"Balance:"+balance);
   }
}
class sav_acc extends account{
   private static double rate=5;
   sav_acc(String name,int acc_no,double balance)
   {
   super(name,acc_no,"savings",balance);
   }
   void interest()
   {
      balance+=balance*(rate)/100;
      System.out.println("balance:"+balance);
   }
```

```
}
class curr_acc extends account{
   curr_acc(String name,int acc_no,double balance)
   {
   super(name,acc_no,"current",balance);
   }
   private double minbal=500;
   private double servicecharge=50;
   void checkmin(){
      if(balance<minbal){
         System.out.println("Insufficient balance so serivce charges are subtracted from your
balance amount");
         balance-=servicecharge;
         System.out.println("Balance is:"+balance);
      }
   }
}
public class bank {
   public static void main(String args[]){
      Scanner s=new Scanner(System.in);
      System.out.println("enter the name :");
      String name=s.next();
      System.out.println("enter the type(current/savings):");
      String type=s.next();
      System.out.println("enter the account number:");
      int acc_no=s.nextInt();
      System.out.println("enter the intial balance:");
      double balance=s.nextDouble();
      int ch;
      double amount1,amount2;
      account acc=new account(name,acc_no,type,balance);
      sav_acc sa=new sav_acc(name,acc_no,balance);
      curr_acc ca=new curr_acc(name,acc_no,balance);
      while(true)
      {
         if(acc.type.equals("savings"))
         {
            System.out.println("\nMenu\n1.deposit 2.withdraw 3.compute interest 4.display");
            System.out.println("enter the choice:");
            ch=s.nextInt();
            switch(ch) {
```

```
            case 1:System.out.println("enter the amount:");
              amount1=s.nextInt();
              sa.deposit(amount1);
              break;
            case 2:System.out.println("enter the amount:");
              amount2=s.nextInt();
              sa.withdraw(amount2);
              break;
            case 3:sa.interest();
              break;
            case 4:sa.display();
              break;
            default:System.out.println("invalid input");
                System.exit(0);
          }
        }
        else{
          System.out.println("\nMenu\n1.deposit 2.withdraw  3.display");
          System.out.println("enter the choice:");
          ch=s.nextInt();
          switch(ch){
            case 1:System.out.println("enter the amount:");
                amount1=s.nextInt();
                ca.deposit(amount1);
                break;
            case 2:System.out.println("enter the amount:");
                amount2=s.nextInt();
                ca.withdraw(amount2);
                ca.checkmin();
                break;
            case 3:ca.display();
                break;
            default:System.out.println("invalid input");
                System.exit(0);

          }
        }
      }
    }
}
```

LAB 06

```
// Student.java
package CIE;
import java.util.Scanner;
public class Student {
   protected String usn = new String();
   protected String name = new String();
   protected int sem;
   public void inputStudentDetails() {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter USN: ");
      usn = scanner.next();
      System.out.print("Enter Name: ");
      name = scanner.next();
      System.out.print("Enter Semester: ");
      sem = scanner.nextInt();
   }
   public void displayStudentDetails() {
      System.out.println("USN: " + usn);
      System.out.println("Name: " + name);
      System.out.println("Semester: " + sem);
   }
}

// Internals.java
package CIE
import java.util.Scanner
public class Internals extends Student {
   protected int marks[] = new int[5];
   public Internals() {
```

```
        // Constructor for Internals
    }
    public void inputCIEmarks() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter Internal Marks for " + name);
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + " marks: ");
            marks[i] = scanner.nextInt();
        }
    }
}
// Externals.java
package SEE;
import CIE.Internals
import java.util.Scanner;
public class Externals extends Internals {
    protected int marks[];
    protected int finalMarks[];
    public Externals() {
        marks = new int[5];
        finalMarks = new int[5];
    }
    public void inputSEEmarks() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter SEE Marks for " + name);
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + " marks: ");
            marks[i] = scanner.nextInt();
        }
    }
```

```
    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++)
            finalMarks[i] = marks[i] / 2 + super.marks[i];
    }
    public void displayFinalMarks() {
        displayStudentDetails();
        for (int i = 0; i < 5; i++)
            System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
    }
}
// Main.java
import SEE.Externals;
public class Main {
    public static void main(String args[]) {
        int numOfStudents = 2;
        Externals finalMarks[] = new Externals[numOfStudents];
        for (int i = 0; i < numOfStudents; 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 < numOfStudents; i++) {
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

LAB 07

```
import java.util.Scanner;
class WrongAge extends Exception {
    public WrongAge(String e) {
        super(e);
    }
}

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

class Father extends InputScanner {
    int fatherAge;

    public Father() throws WrongAge {
        System.out.println("Enter Father's age:");
        fatherAge = s.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
    public void display() {
        System.out.println("Father's age: " + fatherAge);
    }
}
class Son extends Father {
    int sonAge;
    public Son() throws WrongAge {
        super();
```

```
        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 < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

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

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

LAB 08

```
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 09

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
   SwingDemo(){
      // create jframe container
      JFrame jfrm = new JFrame(&amp;quot;Divider App&amp;quot;);
      jfrm.setSize(275, 150);
      jfrm.setLayout(new FlowLayout());
      // to terminate on close
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

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

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

      // calc button
      JButton button = new JButton(&amp;quot;Calculate&amp;quot;);
      // 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(&quot;Action event from a text field&quot;);
        }
      };
      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(&quot;\nA = &quot; + a);
            blab.setText(&quot;\nB = &quot; + b);
            anslab.setText(&quot;\nAns = &quot;+ ans);
          }
          catch(NumberFormatException e){
            alab.setText(&quot;&quot;);
            blab.setText(&quot;&quot;);
            anslab.setText(&quot;&quot;);

            err.setText(&quot;Enter Only Integers!&quot;);
          }
          catch(ArithmeticException e){
            alab.setText(&quot;&quot;);
            blab.setText(&quot;&quot;);
            anslab.setText(&quot;&quot;);
            err.setText(&quot;B should be NON zero!&quot;);
          }
        }
      });// 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

1. Inter process Communication

```
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<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();
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.");
}
}
```

2. Dead Lock

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