| **Name:** | Mahadev Balla |
| --- | --- |
| **UID:** | 2023300010 |
| **Experiment No.** | 10B |

| **AIM:** | Implement programs to demonstrate multi-threading. |
| --- | --- |
| **Program 1** | |
| **PROBLEM STATEMENT :** | Write a program to print "Welcome to SPIT" and "Computer Engineering Department" continuously on the screen in Java using threads. Add a sleep method in the welcome thread to delay its execution for 200ms. |
| **PROGRAM:** | class Welcome extends Thread{    synchronized public void run(){  while(true){  System.out.println("Welcome to SPIT");  CE t2 = new CE();  t2.start();    try {  Thread.sleep(200);  System.out.println();  } catch (Exception e) {  System.out.println(e);  }  }  }  }  class CE extends Thread{  synchronized public void run(){  System.out.println("Computer Engineering Department");  }  }  public class wc {  public static void main(String[] args) {  Welcome t1 = new Welcome();  t1.start();  }  } |
| **RESULT:** | |
| **Program 2** | |
| **PROBLEM STATEMENT :** | Print even numbers by one thread and odd numbers by another thread and print the sum of even and odd numbers by third thread. |
| **PROGRAM:** | import java.util.\*;  class printEven extends Thread {  public int numEven;  static Random random = new Random();  static int randomNumber = random.nextInt(100);  synchronized public void run() {  for (int i = 0; i <= randomNumber; i++) {  if (i % 2 == 0) {  System.out.println("\nEven: " + i);  numEven = i;  }  try {  Thread.sleep(500);  } catch (Exception e) {  }  }  }  }  class printOdd extends Thread {  int number;  public int numOdd;  printOdd(int num) {  this.number = num;  }  synchronized public void run() {  for (int i = 0; i <= number; i++) {  if (i % 2 == 1) {  System.out.println("Odd: " + i);  numOdd = i;  }  try {  Thread.sleep(500);  } catch (Exception e) {  }  }  }  }  class printSum extends Thread {  synchronized public void run() {  printEven Even = new printEven();  printOdd Odd = new printOdd(printEven.randomNumber);  Even.start();  Odd.start();  try {  for (int i = 0; i <= printEven.randomNumber; i++) {  Thread.sleep(1000);  System.out.println("Sum: " + (Even.numEven + Odd.numOdd));  if (Even.numEven == printEven.randomNumber || Odd.numOdd == printEven.randomNumber) {  break;  }  }  } catch (Exception a) {  }  }  }  class numthread {  public static void main(String[] args) {  printSum S = new printSum();  S.start();  }  } |
| **RESULT:** | |
| **CONCLUSION :** Implemented multi-threading in the given problems. | |