**Smt. Chandaben Mohanbhai Patel Institute of Computer Applications**

**Sub: CA314: Object Oriented Programming through JAVA**

**Practical Assignment-9**

**Unit-V**

**Multithreading**

**/\***

**1. Write a program for demonstrating creating thread object using Runnable interface in java.**

**\*/**

**package cmpica;**

**class MyClass implements Runnable {**

**Thread t;**

**public void run()**

**{**

**System.out.println("Thread Started!");**

**}**

**}**

**public class Assignment9\_1 {**

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

**MyClass obj=new MyClass();**

**Thread t1=new Thread(obj);**

**t1.start();**

**}**

**}**

**/\***

**2. Write a program for demonstrating creating thread object using Thread class in java.**

**\*/**

**package cmpica;**

**class MyThread extends Thread{**

**public void run()**

**{**

**System.out.println("Thread Is Running");**

**}**

**}**

**public class Assignment9\_2 {**

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

**MyThread obj=new MyThread();**

**obj.start();**

**}**

**}**

**/\***

**3. Write a program to set the priority of two threads and**

**check which thread executes first and uses more CPU time.**

**\*/**

**package cmpica;**

**class ThreadDemo extends Thread {**

**public void run()**

**{**

**System.out.println("Inside run method");**

**}**

**}**

**public class Assignment9\_3 {**

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

**ThreadDemo t1 = new ThreadDemo();**

**ThreadDemo t2 = new ThreadDemo();**

**ThreadDemo t3 = new ThreadDemo();**

**System.out.println("t1 thread priority : "+ t1.getPriority());**

**System.out.println("t2 thread priority : "+ t2.getPriority());**

**System.out.println("t3 thread priority : "+ t3.getPriority());**

**t1.setPriority(2);**

**t2.setPriority(5);**

**t3.setPriority(10);**

**System.out.println("t1 thread priority : "+ t1.getPriority());**

**System.out.println("t2 thread priority : "+ t2.getPriority());**

**System.out.println("t3 thread priority : "+ t3.getPriority());**

**}**

**}**

**/\***

**4. Create an Account class having methoddeposit() & withdraw() methods.**

**Make use of synchronization while withdrawing money in case of joint account.**

**\*/**

**package cmpica;**

**class Account {**

**int balance= 1000;**

**public int getBal(){**

**return balance;**

**}**

**public void withdraw(int bal){**

**balance= balance-bal;**

**}**

**public void Methoddeposit(int bal){**

**balance= balance+bal;**

**}**

**}**

**class ThreadLogic implements Runnable**

**{**

**Account acc=new Account();**

**public void run()**

**{**

**for (int i = 0; i < 3; i++) {**

**makeWithdraw(100);**

**if (acc.getBal() < 0) {**

**System.out.println("account is overdrawn!");**

**}**

**deposit(200);**

**}**

**}**

**private synchronized void makeWithdraw(int bal){**

**if (acc.getBal()>=bal) {**

**System.out.println(Thread.currentThread().getName()+" "+ "is try to withdraw");**

**try {**

**Thread.sleep(100);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**acc.withdraw(bal);**

**System.out.println(Thread.currentThread().getName()+" "+ "completed the withdraw");**

**}else{**

**System.out.println(Thread.currentThread().getName()+ " "+"doesn't have enough money for withdraw ");**

**}**

**}**

**private synchronized void deposit(int bal){**

**if (bal>0) {**

**System.out.println(Thread.currentThread().getName()+" "+ " is try to deposit");**

**try {**

**Thread.sleep(100);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**acc.Methoddeposit(bal);**

**System.out.println(Thread.currentThread().getName()+" "+ "completed the deposit");**

**}else{**

**System.out.println(Thread.currentThread().getName()+ " "+"doesn't have enough money for deposit");**

**}**

**}**

**}**

**public class Assignment9\_4 {**

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

**ThreadLogic ts = new ThreadLogic();**

**Thread t1 = new Thread(ts, "Partner 1");**

**Thread t2 = new Thread(ts, "Partner 2");**

**Thread t3 = new Thread(ts, "Partner 3");**

**t1.start();**

**t2.start();**

**t3.start();**

**}**

**}**

**/\***

**5. Create two thread, first thread will print prime number and second thread will print perfect number between 1-100.**

**\*/**

**package cmpica;**

**class FirstThread extends Thread{**

**public void run()**

**{**

**int i =0;**

**int num =0;**

**String primeNumbers = "";**

**for (i = 1; i <= 100; i++)**

**{**

**int counter=0;**

**for(num =i; num>=1; num--)**

**{**

**if(i%num==0)**

**{**

**counter = counter + 1;**

**}**

**}**

**if (counter ==2)**

**{**

**primeNumbers = primeNumbers + i + " ";**

**}**

**}**

**System.out.println("\n\nPrime numbers from 1 to 100 are :");**

**System.out.println(primeNumbers);**

**}**

**}**

**class SecondThread extends Thread{**

**public void run()**

**{**

**int i,sum=1;**

**System.out.print("Perfect Numbers are: 1,");**

**for(int j=2;j<=100;j++)**

**{**

**for(i=2;i<j;i++)**

**{ if(j%i==0)**

**sum=sum+i;**

**}**

**if(j==sum)**

**System.out.print(j+",");**

**}**

**System.out.println();**

**}**

**}**

**public class Assignment9\_5 {**

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

**FirstThread ft=new FirstThread();**

**SecondThread st=new SecondThread();**

**ft.start();**

**st.start();**

**}**

**}**

**/\***

**6. Write a java program to demonstrate the implementation of Synchronization in java.**

**\*/**

**package cmpica;**

**class Question6**

**{**

**void printTable(int n)**

**{**

**for(int i=1;i<=5;i++)**

**{**

**System.out.println(n\*i);**

**try**

**{**

**Thread.sleep(400);**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**}**

**}**

**class MyThread1 extends Thread**

**{**

**Question6 obj;**

**MyThread1(Question6 t)**

**{**

**this.obj=t;**

**}**

**public void run()**

**{**

**obj.printTable(5);**

**}**

**}**

**class MyThread2 extends Thread**

**{**

**Question6 obj;**

**MyThread2(Question6 t)**

**{**

**this.obj=t;**

**}**

**public void run()**

**{**

**obj.printTable(100);**

**}**

**}**

**public class Assignment9\_6 {**

**public static void main(String args[])**

**{**

**Question6 obj = new Question6();**

**MyThread1 t1=new MyThread1(obj);**

**MyThread2 t2=new MyThread2(obj);**

**t1.start();**

**t2.start();**

**}**

**}**

**/\***

**7. Write a java program to demonstrate the implementation of inter thread communication in java.**

**\*/**

**package cmpica;**

**class ThreadB extends Thread {**

**int totalBalance = 0;**

**public void run() {**

**synchronized (this) {**

**System.out.println("child Thread starts calculation for total balance");**

**for (int i = 0; i <= 50; i++) {**

**totalBalance = totalBalance + i;**

**}**

**System.out.println("child thread gives notification call");**

**this.notify();**

**}**

**}**

**}**

**public class Assignment9\_7 {**

**public static void main(String[] args) throws InterruptedException {**

**ThreadB b = new ThreadB();**

**b.start();**

**synchronized (b) {**

**System.out.println("main thread calling wait() method");**

**b.wait();**

**System.out.println("main thread got notification call");**

**System.out.println("totol balance " + b.totalBalance);**

**}**

**}**

**}**

**/\***

**8. Write a java program to print from 1-15 and 15-1 using multithreading in java.**

**\*/**

**package cmpica;**

**class Thread1 implements Runnable {**

**public synchronized void run() {**

**print1();**

**print2();**

**}**

**public void print1() {**

**for (int i = 0; i <= 15; i += 1) {**

**System.out.println(i+" "+Thread.currentThread());**

**}**

**}**

**public void print2() {**

**for (int i = 15; i >=1; i--) {**

**System.out.println(i+" "+Thread.currentThread());**

**}**

**}**

**}**

**public class Assignment9\_8 {**

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

**Thread1 t1 = new Thread1();**

**Thread td1 = new Thread(t1);**

**Thread td2 = new Thread(t1);**

**td1.start();**

**td2.start();**

**}**

**}**

**/\***

**9. Write a program to reverse the digits using threading in java.**

**\*/**

**package cmpica;**

**class Secondlast implements Runnable{**

**public void run()**

**{**

**reverse();**

**}**

**void reverse()**

**{**

**int number = 987957, reverse = 0;**

**while(number != 0)**

**{**

**int remainder = number % 10;**

**reverse = reverse \* 10 + remainder;**

**number = number/10;**

**}**

**System.out.println("The reverse of the given number is: " + reverse);**

**}**

**}**

**public class Assignment9\_9 {**

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

**Secondlast sl=new Secondlast();**

**Thread th=new Thread(sl);**

**th.start();**

**}**

**}**

**/\***

**10. Write a program to print odd and even number from 1-20 using threading.**

**\*/**

**package cmpica;**

**class PrintNumbers extends Thread{**

**public void run()**

**{**

**PrintOdd();**

**PrintEven();**

**}**

**void PrintOdd()**

**{**

**System.out.println("Odd Numbers Are: ");**

**for (int i=1;i<=20;i=i+2)**

**{**

**System.out.println(i);**

**}**

**}**

**void PrintEven()**

**{**

**System.out.println("\n\nEven Numbers Are: ");**

**for (int i=2;i<=20;i=i+2)**

**{**

**System.out.println(i);**

**}**

**}**

**}**

**public class Assignment9\_10 {**

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

**PrintNumbers pn=new PrintNumbers();**

**pn.start();**

**}**

**}**