

Smt. Chandibai Himathmal Mansukhani College

Contents

USCS3P01:USCS303-Operating System (OS) Practical-05	2
Threads	2
Practical Date: 14th August,2021	2
Practical Aim: Threads(Multi-Threading)	2
Thread States: Life Cycle of a Threads : -	2
.....	2
1. New and Runnable States :	3
2. Waiting State:.....	3
3. Timed Waiting State:.....	3
4. Blocked State:.....	3
5. Terminated State:	3
Summation.....	4
Question-01:.....	4
Source Code:.....	4
Output:.....	6
Primes.....	6
Question-02:.....	6
Source Code 1:.....	6
Source Code 2:.....	8
Output:.....	10
Fibonacci.....	11
Question-03:.....	11
Source Code:.....	11
Output :	13

Smt. Chandibai Himathmal Mansukhani College

USCS3P01:USCS303-Operating System (OS) Practical-05

Threads

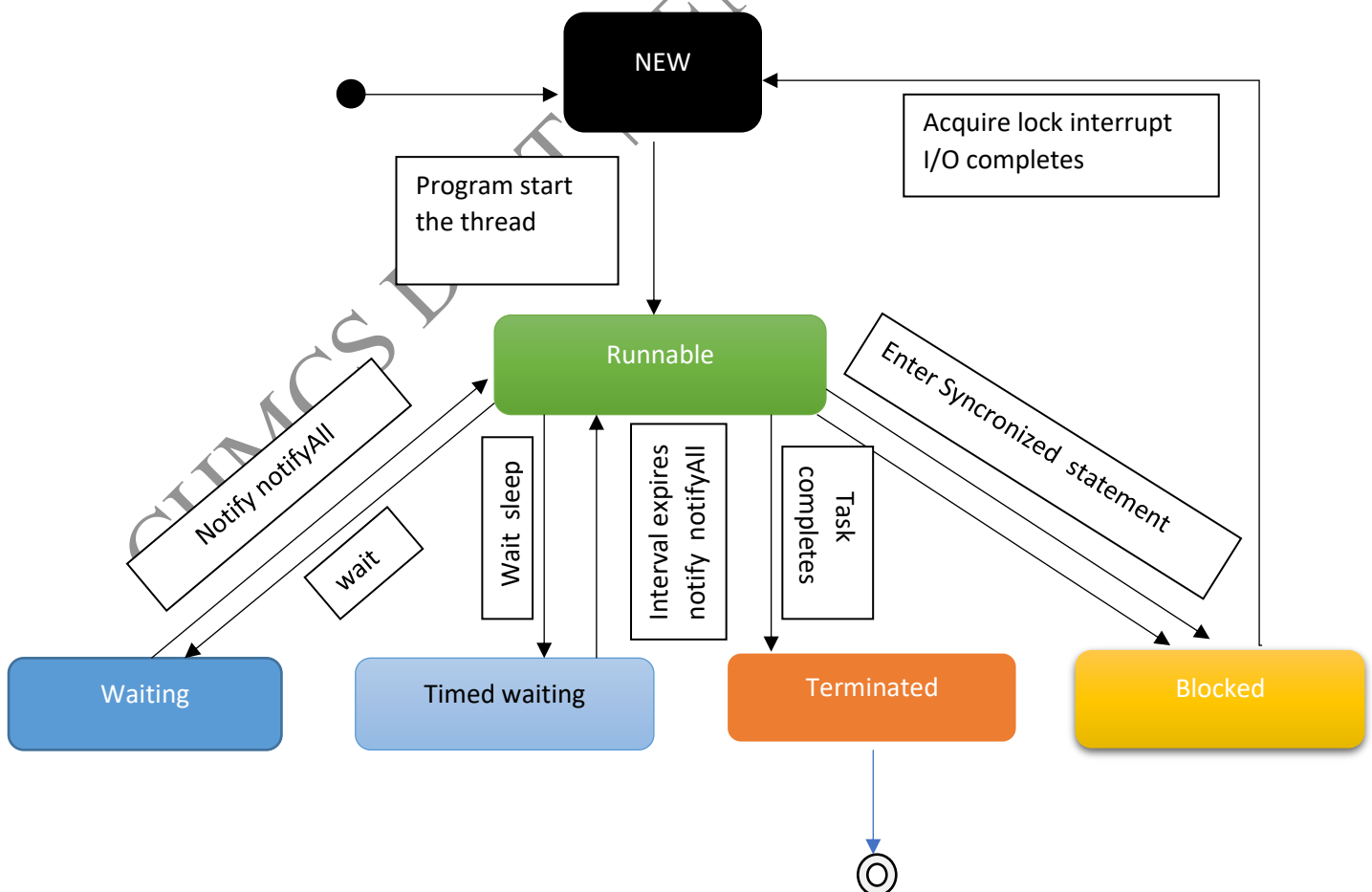
Practical Date: 13th August,2021

Practical Aim: Threads(Multi-Threading)

Thread States: Life Cycle of a Threads :-

A java thread can be in any of following thread states during its life cycle i.e.

- New,
- Runnable,
- Blocked,
- Waiting,
- Timed Waiting or Terminated.



Smt. Chandibai Himathmal Mansukhani College

1. New and Runnable States :

- A new thread begins its life cycle in the new state.
- It remains in this state until the program starts the thread , which places in the running state.
- A thread in the runnable state is considered to be excuting its task.

2. Waiting State:

- Sometimes a runnable thread transition to the waiting state while it waits for another thread to perform a task.
- A waiting thread transition back to the runnable state only when another thread notifies it to continue executing .

3. Timed Waiting State:

- A runnable thread can enter the timed waiting state for a specified interval of time . It transition back to the runnable state when the time interval expires or when the event it's waiting for occurs .

4. Blocked State:

- A runnable thread transition to the blocked state when it attempts to perform a task that cannot be complete immediately and it must temporarily wait until the task completes.

5. Terminated State:

- A runnable thread enters the terminated state (sometimes called dead state) when it successfully completes its task or otherwise terminates (perhaps due to an error).

Smt. Chandibai Himathmal Mansukhani College

Summation

Question-01:

Write a multithreaded java program that determines the summation of a non -negative integer. The Summation class implements the Runnable interface . Thread creation is performed by creating an object instance of the Thread class and passing the constructor a Runnable object.

Source Code:

//Name: Yash Patil

// Batch: B2

// PRN: 2020016400809191

// Date: 13-08-2021

// Prac-05: Threads

```
class P5_Q1_Summation_YP implements Runnable
{
    int upperLimit,sum;
    public P5_Q1_Summation_YP (int upperLimit)
    {
        this.upperLimit=upperLimit;
    }
    public void run()
    {
        for(int i =1;i<=upperLimit;i++)
            sum +=i;
    }
}
//ends of class P5_Q1_Summation_YP
public class P5_Q1_SummationTest_YP
{
}
```

Smt. Chandibai Himathmal Mansukhani College

```
public static void main(String args[])
{
    if(args.length<= 0)
        System.out.println("Usage:
P5_Q1_SummationTest_YP<integervalue>");
    else
    {
        int upp = Integer.parseInt(args[0]);
        if(upp<=0)
            System.out.println("args[0]: " + args[0] + " must be a positive
number");
        else
        {
            P5_Q1_Summation_YP s = new P5_Q1_Summation_YP(upp);
            Thread t = new Thread(s);
            t.start();
            try{
                t.join();
                System.out.println("The sum of first " + upp + "
elements is " + (s.sum));
            }
            catch(Exception e){
                e.printStackTrace();
            }
        }
    }
}
//outer else ends
}
//main ends
}
//end of class class P5_Q1_SummationTest_YP
```

Smt. Chandibai Himathmal Mansukhani College

Output:

Primes

Question-02:

Write a multithreaded java program that outputs prime numbers. This program should work as follows :

The user will run the program and will enter a number on the command line. The program will then create a separate thread that outputs all the prime numbers less than or equal to the numbers entered by the user.

Source Code 1:

```
//Name: Yash Patil
```

```
// Batch: B1
```

```
// PRN: 2020016400809191
```

Smt. Chandibai Himathmal Mansukhani College

// Date: 13-08-2021

// Prac-05: Threads

```
import java.io.*;
import java.util.*;

public class P5_Q2_Primes_YP {
    public static void main(String args[]){
        try{
            P5_Q2_PrimeThread_YP pt = null;
            System.out.print("Enter a number> ");
            Scanner scan = new Scanner(System.in);
            int limit = scan.nextInt();
            System.out.print("Enter a file name to store the results>");
            String fName = scan.next();
            if(fName.length()>0)
                pt = new P5_Q2_PrimeThread_YP(limit, new
FileOutputStream(fName));
            else
                pt = new P5_Q2_PrimeThread_YP(limit);
            pt.run();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
} //main ends
} //class ends
```

Smt. Chandibai Himathmal Mansukhani College

Source Code 2:

//Name: Yash Patil

// Batch: B2

// PRN: 2020016400809191

// Date: 13-08-2021

// Prac-05: Threads

```
import java.io.*;
```

```
class P5_Q2_PrimeThread_YP extends Thread {
```

```
    private PrintStream pOut = null;
```

```
    private int limit = 0;
```

```
    //default constructor.does nothing
```

```
    public P5_Q2_PrimeThread_YP(){
```

```
    }
```

```
//constructor to set the number below which to generate primes
```

```
//no output stream is specified,so it outputs to the System.out
```

```
    public P5_Q2_PrimeThread_YP(int I){
```

```
        limit = I;
```

```
        try{
```

```
            pOut = System.out;
```

```
        }catch(Exception e){
```

```
            e.printStackTrace();
```

```
        }
```

```
    }
```

```
//constructor that sets both the number, as above, and specifies an output stream
```

```
//if the specified stream is null, uses System.out
```

```
    public P5_Q2_PrimeThread_YP(int I, OutputStream outS){
```

```
        limit = I;
```


Smt. Chandibai Himathmal Mansukhani College

```
try{
    if(outS != null){
        pOut = new PrintStream(outS);
    }else{
        pOut = System.out;
    }
} catch(Exception e){
    e.printStackTrace();
}

}

//method that performs the work of the thread,
//in this case the generation of prime numbers.
public void run(){
    //compute primes via the seive
    boolean numbers[] = new boolean[limit+1];
    numbers[0] = false;
    numbers[1] = false;
    for(int i = 2; i<numbers.length; i++){
        numbers[i] = true;
    }
    for(int i = 2; i<numbers.length; i++){
        if(numbers[i]){
            for(int j=(2*i);j< numbers.length;j+=i){
                numbers[j] = false;
            }
        }
    }
    //inner for ends
    //if ends
    //outer for ends
    for(int i=0;i< numbers.length;i++){
        if(numbers[i])
            pOut.println(i);
    }
}
```

Smt. Chandibai Himathmal Mansukhani College

}//for ends

}//run ends

}//class ends

Output:

CHMCS DEPT / SEM-III / 2021-2022

Smt. Chandibai Himathmal Mansukhani College

Fibonacci

Question-03:

The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5, 8,Formally, it can be expressed as : $fib_0 = 0$, $fib_1 = 1$, $fib_n = fib_{n-1} + fib_{n-2}$. Write a multithreaded program that generates the Fibonacci sequence using either the Java.

Source Code:

```
//Name: Yash Patil
```

```
// Batch: B1
```

```
// PRN: 2020016400809191
```

```
// Date: 13-08-2021
```

```
// Prac-05: Threads
```

```
import java.util.ArrayList;
```

```
import java.util.Scanner;
```

```
public class P5_Q3_Fibo_YP
```

```
{
```

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

```
        Scanner scan = new Scanner(System.in);
```

```
        ArrayList al = new ArrayList();
```

```
        int a;
```

```
        System.out.print("Enter the number: ");
```

```
        a = scan.nextInt();
```

```
        P5_Q3_FiboThread_YP fibTh = new P5_Q3_FiboThread_YP(a);
```

```
        fibTh.start();
```

```
        try{
```

```
            fibTh.join();
```

```

    }catch(InterruptedException ex){
        ex.printStackTrace();
    }

    int fseries[] = fibTh.arr;

    System.out.println("First "+a+" fibonacc numbers are:");

    for(int i=0;i<a;i++){

        System.out.print(fseries[i]+ " ");

    }

} //main ends

} //class ends

class P5_Q3_FiboThread_YP extends Thread
{

    private int a,i;

    Thread t;

    int arr[];

    public P5_Q3_FiboThread_YP(int a){

        this.a = a;

        arr = new int[a];

    }

    public void run(){

        arr[0] = 0;

        arr[1] = 1;

        for(i=2;i<a;i++){

            arr[i] = arr[i-1] + arr[i-2];

        }

    } //run ends

} //class ends

```

Smt. Chandibai Himathmal Mansukhani College

Output :

CHMCS DEPT / SEM-III / 2021-2022