PRESIDENCY UNIVERSITY

(Established under the Presidency University Act, 2013 of the Karnataka Act 41 of 2013)

CSE3146 – Advanced JAVA Programming

LAB SHEET - 1

Module 1- Multithreading using Java

Q1. Create a single thread by implementing Runnable interface.

Solution:

Step 1: Start

Step 2: Create a class Task that implements builtin interface Runnable

Step 3: Override run() to define work of thread. Use Thread.CurrentThread() to print the thread is running currently

Step 4: Create object task of Task.

Step 5: Create object of Thread by passing task as parameter

Step 5: Assign a name to thread using setName("Name")

Step 6: Call start() to start the thread

Step 7: Stop

```
import java.lang.*; //optional
class Task implements Runnable {
    public void run() {
       System.out.println(Thread.currentThread()+" printing ");
        System.out.println("Welcome");
    }
}
  public class TestThread {
  public static void main(String[] args) {
  Task task = new Task();
  Thread t1= new Thread(task);
   t1.setName("first");
   t1.start();
 System.out.println(Thread.currentThread()+" printing ");
 System.out.println("to Java");
  }
}
```

Q2. Create a single thread by extending Thread class

Solution:

Step 1: Start

Step 2: Create a class MyThread that extends builtin class Thread

Step 3: Override run() to define work of thread. Use Thread.CurrentThread() to print the thread is running currently

Step 4: Create object of MyThread

Step 5: Assign a name to thread using setName("Name")

Step 6: Call start() to start the thread

Step 7: Stop

```
class MyThread extends Thread {
     // run() method to perform action for thread.
     public void run()
     {
        int a= 10;
        int b=12;
        int result = a+b;
        System.out.println(Thread.currentThread()+" started
running..");
        System.out.println("Sum of two numbers is: "+ result);
       System.out.println(Thread.currentThread()+" completed..");
     }
}
public class TestThread {
     public static void main( String args[] )
     {
     System.out.println(Thread.currentThread()+" started");
      // Creating instance of the class extend Thread class
        MyThread t = new  MyThread();
        t.setName("first");
        //calling start method to execute the run() method of the
Thread class
        t.start();
     System.out.println(Thread.currentThread()+" completed");
     }
}
```

Q3: Create 3 threads 1st, 2nd and 3rd to print numbers 5 to 1 concurrently by extending Thread Class.

Requirement:

- Override run() to print 5 to 1 using for loop
- Use sleep() for switching the context to other threads
- Use setName() to set the name of Thread or Use constructor Thread() to set the name of Thread

Sol:

```
class MyThread extends Thread {
String name;
        MyThread (String name){
               setName(name);   or //  super(name);
               this.name=name;
            System.out.println( "A New thread: " + name + "is created\n" );
    }
    public void run() {
    try {
        for(int j = 5; j > 0; j--) {
            System.out.println(name + ": " + j);
            Thread.sleep(1000);
        }
    }catch (InterruptedException e) {
        System.out.println(name + " thread Interrupted");
    }
     System.out.println(name + " thread exiting.");
    }
}
public class TestMultiThread {
    public static void main(String args[]) {
        MyThread t1=new MyThread("one");
        MyThread t2=new MyThread("two");
        MyThread t3=new MyThread("three");
        t1.start();
        t2.start();
        t3.start();
        try {
            Thread.sleep(8000);
        } catch (InterruptedException excetion) {
            System.out.println("Inturruption occurs in Main Thread");
        }
        System.out.println("We are exiting from Main Thread");
    }
```

Q4: Create 3 threads 1st, 2nd and 3rd to print factorial of three different numbers concurrently by extending Thread Class.

Requirement:

- Override run() to print factorial using for loop
- Use sleep() for switching the context to other threads
- Use constructor Thread() to set the name of Thread
- Demonstrate join() and isAlive() method

Sol:

```
class MyThread extends Thread {
String name;
int number;
long fact=1;
        MyThread (int number,String name){
               super(name);    //calling Thread()
               this.number=number;
               this.name=name;
              System.out.println( "A New thread: " + name + " is created\n" );
    }
    public void run() {
    try {
        for(int i = 1; i <= number; i++) {
            System.out.println(name + " calculating factorial");
            fact=fact*i;
            Thread.sleep(1000);
        }

    }catch (InterruptedException e) {
        System.out.println(name + " thread Interrupted");
    }
     System.out.println(name + " calculated factorial "+fact);
    }  }
public class TestMultiThread {
    public static void main(String args[]) {
        MyThread t1=new MyThread(5,"one");
        MyThread t2=new MyThread(4,"two");
        MyThread t3=new MyThread(3,"three");
        t1.start();
        t2.start();
        t3.start();
     System.out.println("1st Alive : "+t1.isAlive());
     System.out.println("2nd Alive : "+t2.isAlive());
     System.out.println("3rd Alive : "+t3.isAlive());
        try {
          t1.join();
           t2.join();
           t3.join();
System.out.println("1st Alive : "+t1.isAlive());
System.out.println("2nd Alive : "+t2.isAlive());
System.out.println("3rd Alive : "+t3.isAlive());
} catch (InterruptedException excetion) {
            System.out.println("Inturruption occurs in Main Thread");
        }
```

```
        System.out.println("We are exiting from Main Thread");
    }
}
```

Q5: Create three threads by setting different priorities to each thread.

Sol:

```
class ThreadPrior extends Thread {

    public void run()
    {
        // Print statement
        System.out.println("Inside run method");
    }

}
public class TestThreadPrior {
    public static void main(String[] args)
    {
        ThreadPrior t1 = new ThreadPrior();
        ThreadPrior t2 = new ThreadPrior();
        ThreadPrior t3 = new ThreadPrior();

        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(8);

        t3.setPriority(21); //error
        System.out.println("t1 thread priority : " + t1.getPriority());
        System.out.println("t2 thread priority : "+ t2.getPriority());

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

        // Main thread

        System.out.println("Currently Executing Thread :
"+Thread.currentThread().getName());

        System.out.println(
            "Main thread priority : "+ Thread.currentThread().getPriority());

        // Main thread priority is set to 10
        Thread.currentThread().setPriority(10);

        System.out.println(
            "Main thread priority : "+ Thread.currentThread().getPriority());
    }
}
```

Q6: Demonstrate Thread Synchronization for a given resource to avoid race condition.

- Create a Resource class to keep two resources [and]. No thread can take] without [
- Create three threads to access the above resource without synchronization
- Access the above resource using synchronization

Sol:

```
class Resource {
     void use(String name) {
     System.out.print("[" + name);
     try {
      Thread.sleep(1000);
   } catch(InterruptedException e) {
         System.out.println("Interrupted");
      }
     System.out.println("]");
     }
}

class MyThread extends Thread {
       String name;
       Resource r;
        MyThread (String name,Resource r){
        super(name);
        this.name = name;
        this.r=r;
    }
    public void run() {
        synchronized(r) {
        r.use(name);
       }
    }
}
public class TestMultiThread {
    public static void main(String args[]) {
        Resource res=new Resource();
        MyThread t1=new MyThread("1st",res);
        MyThread t2=new MyThread("2nd",res);
        MyThread t3=new MyThread("3rd",res);
        t1.start();
        t2.start();
        t3.start();
        try {
            t1.join();
            t2.join();
            t3.join();
        } catch (InterruptedException excetion) {
            System.out.println("Inturruption occurs in Main Thread");
        }
        }
}
```

Record Writing

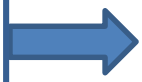
Note: Submit the record on or before the due date.

Both Soft and Hard copies are required to be submitted.

RECORD WRITING INSTRUCTIONS

1. Solve the programming exercise using any IDE (Laptop / Mobile) or using any online compiler.
   A. Students can use online compiler or any preferable platform for the execution. Suggested is to use JDoodle. https://www.jdoodle.com/online-javacompiler/ Do test this site before your CA.
   B. Mobile users, kindly install JStudio - ide for java https://play.google.com/store/apps/details?id=com.qamar.ide.java&hl=en. This instruction is already given for solving your lab programs. If you haven't done, please do install, and test the app as soon as possible.
2. While solving your programming exercise, write the code in A4 sheet paper/Record. While writing on the paper, please add these info. "Presidency University" "Department of CSE" "Odd semester 2021-2022" "MODULE - 1" "Course code : CSA 1005", Course name : OOP Using JAVA, ID:_____, NAME:_____, SEC:____ , Date:_____
   A. While coding (the soft copy) & writing in the paper/record, all your **CLASS NAME** and the **METHOD NAME** must be appended with your **LAST FOUR DIGIT student ID**. This is mandatory, even while WRITING in the paper/record.

For example: If your Registration number is 2020BCA0161 then



```
class sample0161   {
        void Method0161(parameterlist)  {
                //method body
        }
```

   B. Take a screenshot of **your program & the output** from your mobile/laptop.
   C. Take a photo of the handwritten program.
   D. Put together (**4. A,B,C**) , **combine as one pdf**, with the file name as your student **registration number(ex. 2020BCA0161.pdf)**, and upload the file in Camu.
   E. The document must be uploaded within the specified time in Camu.

Kindly follow the instructions very carefully so that your submission will be valid.