FPT-Aptech

Java Programming II

Module 1: Introduction to Threads
Lab Guide for Lab1

Session Objectives

In this session, you will be practicing with

- Creating Threads
- Thread States
- Methods of Thread class
- Managing Threads
- Daemon Threads

Part 1 – Getting started (30 minutes)

1. Create new thread class by extending the class **Thread**

The following program simulates a car racing using multi-threading technique. RaceCar extends Thread, therefore, it is runnable. In the Race class, an array of RaceCar objects is created and started asynchronously.
Scan the code first, type the code, compile, run and observe the result.

```
//RaceCar method extends Thread. Therefore, It is runnable.
class RaceCar extends Thread
{
      int finish;
      String name;

      RaceCar(int finish, String name){
            this.finish = finish;
            this.name = name;
      }
   //Override run() method for racing logic
      public void run(){
            for (int i=0; i<finish; i++){
                  System.out.println(name + ": " + (i+1) +
"running…");
```

```
                try {
     //sleep for a random amount of time. To get a random
value,
     //java.util.Math.random() method is used.

Thread.sleep(Math.round(Math.random()*5000));
                } catch(Exception e){}
          }

          System.out.println(name + " finished");
     }
}

public class Race
{
     public static void main(String a[]){
     //create an array of runnable RaceCar objects
          RaceCar[] cars = new RaceCar[5];
          cars[0] = new RaceCar(10, "Mario");
          cars[1] = new RaceCar(10, "Songoku");
          cars[2] = new RaceCar(10, "Herman");
          cars[3] = new RaceCar(10, "Doremon");
          cars[4] = new RaceCar(10, "Hoang Phi Hong");
     //start racing
          for(int i=0; i<5; i++)
               cars[i].start();
     }
}
```

Part 2 – Workshops (30 minutes)

- Quickly look at Module 1's workshops for reviewing basic steps for creating and managing threads.
- Try to compile, run and observe the output of sample code provided for related workshop. Discuss with your class-mate and your instructor if needed.

Part 3 – Lab Assignment (60 minutes)

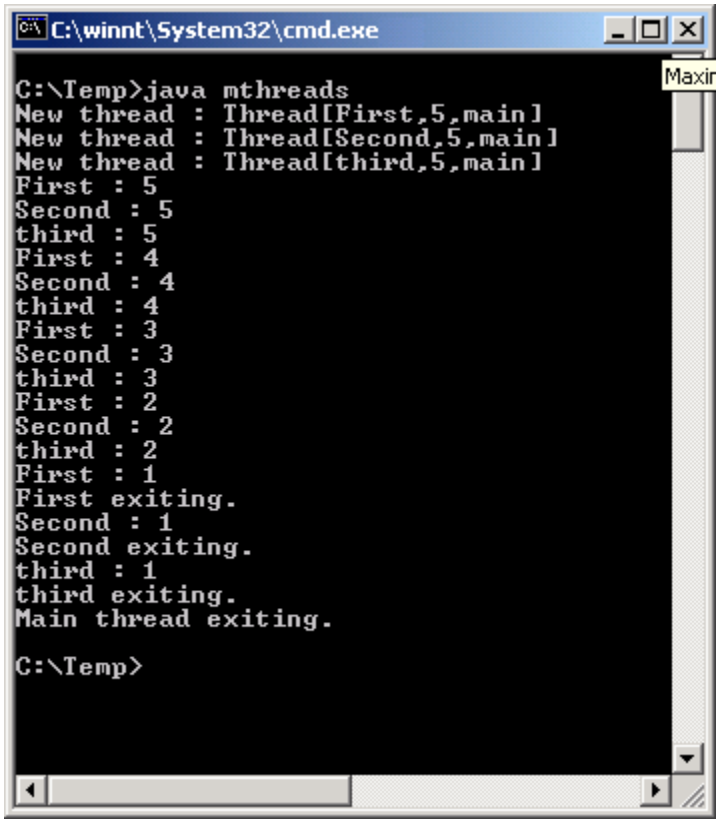
Do the assignment for Module 1 carefully. Discuss with your class-mates and your instructor if needed.

Part 4 – Do it yourself

1.1. Create a Thread named MyThread (by extending from the class Thread or by implementing the interface Runnable) does these tasks:

- Display name of the current thread.
- Rename the created thread to myJavaThread and display this name.
- Overwrite the method run() by displaying the first 10 even numbers. Delay time between each display time is 1500 ms.

1.2. Write a program to create the output as following figure:



Create three threads and the main thread. Execute each thread as simultaneous tasks. Display information when exiting each thread.

1.3. Create a new thread class that prints a value in duration. Information about a thread includes: message, duration, priority.

```
MESSAGE BOARD
====================
Number of messages: 2
Message 1: multithreading
Timeout: 1000
```

```
Priority: high
Message 2: multitasking
Timeout: 2000
Priority: medium
Result:
Multithreading
Multithreading
Multitasking
Multithreading
Multithreading
Multitasking
…..
```