# Department of Computing

**CS213: Advanced Programming**

**Class: BSCS – 6C**

# Lab 3: Development of a Multithread Application

**Date: September 28, 2018**

# Time: Friday 2:00 PM – 5:00 PM

# Instructor: Dr. Abdul Ghafoor

# 

# Lab 3: Development of Multithread Application

## Introduction

In this lab, students have to design, develop, and test a multithreaded application which will read a configuration file to get number of components and sleep time of the thread. The application will dynamically add *button* component in the frame. The total number of components depends on the value given in the configuration file. Each component will change its background color randomly after expiration of the sleep time given in the configuration file. Furthermore, each component must be handled by a separate thread.

## Objectives

* Develop a multithreaded application
* Handling of configuration file using singleton pattern.
* Exception Handling

## Tools/Software Requirement

* Solutions should be made using Java.
* **Do not use any external library for reading configurations.**

**Description**

Each student must, individually build the complete application on their own. Students must upload their solutions on LMS to qualify for evaluation.

* Any exceptions or errors leading to non-execution of submitted code.
* Failure to upload the solution on LMS.
* Failure to submit original code.
* Failure to explain the submission, during viva.

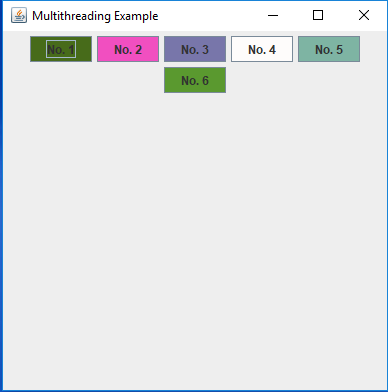
**Lab Task**

Develop a java based multithreaded application in which your application will accept configuration file from command line. The property file has following attributes and format:

*falshTime=1000*

*noOfItems=6*

Your application will accept this file and then will extract the value of *falshTime* and *noOfItems*. If configuration values are not correct then your application must display error message and provides sufficient value for debugging. *noOfItems* of property indicates the total number of components those should be displayed on the GUI as shown in the following diagram.

**

Each component on the GUI is managed by a separate thread so during its execution, the owner thread is responsible to change its background color randomly and then wait for *falshTime (ms)* before changing next background color. The same behavior will be repeated until the user closes the application.

## Deliverables

* Each submission is individual with the following composition:
  + Source Code
  + README.txt (Introduction, Approach, How to Run)
* Convert your submission files into a zip folder and name it as given below, finally upload the zip folder to LMS.
  + Name – Registration No. – Section

## Grade Criteria

This lab is graded. Min marks: 0. Max marks: 10.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Minimum** | **Maximum** |
| Documentation with clearly defined understanding of the lab task and approach | 0 | 2 |
| Code clarity with clean, formatted and commented code. | 0 | 3 |
| Functionality | 0 | 3 |
| Viva | 0 | 2 |
| **Total** | **0** | **10** |