Experiment – 9

1. **Aim:** To implement the concept of thread management and synchronization using concurrent programming

2. **Objective:** After performing the experiment, the students will be able to implement

- Understand the concept of process and thread
- Create a thread
- Perform simple thread operations

3. **Lab objective mapped:** To **understand** alternative paradigm through concurrent programming fundamentals and **design**, **develop** applications based on concurrent programming (PSO2) (PO2)

4. **Prerequisite:** Basics of Java programming- classes, objects, functions, data abstraction

5. **Requirements:** The following are the requirements – Java (JDK8) Compiler

6. **Pre-Experiment Theory:**

Concurrency generally refers to events or circumstances that are happening or existing at the same time. In programming terms, concurrent programming is a technique in which
- Two or more processes start
- Run in an interleaved fashion through switching and
- Complete in an overlapping time period by managing access to shared resources

Process
- Process means any program is in execution.
- Process control block contains information about processes for example Process priority, process id, process state, CPU, register, etc.
- A process can create other processes which are known as **Child Processes**.
- Process takes more time to terminate and it is isolated means it does not share memory with any other process.
- Process is called heavy-weight process **Thread**
- One or more **threads** run in the context of the **process**.
- A **thread** is the basic unit to which the operating system allocates processor time.
- A **thread** can execute any part of the **process** code, including parts currently being executed by another **thread**
- ☐ **Thread is called light weight process**

7. **Laboratory Exercise**
A. Steps to be implemented

- ☐ 1. **Compilation using JDK 8 using Turbo C**
- Write the code in notepad and save as .java file.

- Run command prompt and set the path (Eg. Set path= 'C:\Users\m09mu\Desktop\Javacodes')
- Compile the code using the command 'javac name_of_file.java' □ Correct compile time errors (if any) and rerun the code

1

- After successful compilation, run the code using the command 'java name_of_file'

□ **Using Online IDE for C/C++/Java**
- Log on to www.onlinegdb.com,/ www.jdoodle.com
- Select the programming language for coding
- Create new project
- Save the project using CTRL+S
- Run the program using F9
- For debugging use F8, along with step-into function of onlinegdb

B. Program Code
1. Create a thread t1 to display message "Hello World".
2. Create two threads t1 and t2 to display message "Welcome to SFIT" and "Welcome to IT".
3. Create a thread t1 by implementing runnable interface to display "Hello world" message 4. WAP to display n-thread ids

8. Post Experimental Exercise-

A. Questions:
1. Define Process and highlight its important points
2. Define thread and highlight its important points
3. Explain the life cycle of a thread with proper diagram.

B. Results/Observations/Program output:
Present the program input/output results if any and comment on the same.

C. Conclusion:
1. Write what was performed in the experiment
2. Write which tools you used to perform the experiment
3. Write what you inferred from the output obtained

D. References:
[1] Michael L Scott, "Programming Language Pragmatics", Third edition, Elsevier publication [2] Doug Lea, "Concurrent Programming in Java: Design Principles and Pattern