Page No.
Date

Experiment - 6

Aim: Multithreading
(a). Create thread using Thread class & Runnable Interface
(b). Synchronized keyword

Software Used : Netbeans IDE.

Theory

Multithreading in Java is a feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. Each part of such program is called a thread. So, threads are light-weight processes within a process.

Threads can be created by using two mechanisms :
1. Extending the Thread class.
2. Implementing the Runnable Interface.

Thread Creation by extending the Thread class
We create a class that extends the java.lang.Thread class. This class overrides the run() method available in the Thread class. A thread begins its life inside run() method. We create an object of our class & call start() method to start the execution of thread.

Page No.
Date

Thread creation by implementing the Runnable Interface
We create a new class which implements java.lang.Runnable interface & override run() method. Then we instantiate a Thread object & call start() method on this object.

Synchronized in Java

Multi-threaded programs may often come to a situation where multiple threads try to access the same resources & finally produce erroneous and unforeseen results.

Thus, Java provides a way of creating threads and synchronizing their task by using synchronized blocks. Synchronized blocks in Java are marked with the synchronized keyword. A synchronized block in Java is synchronized on some object. All synchronized blocks synchronized on the same object can only have one thread executing inside them at a time. All other threads attempting to enter the synchronized block are blocked until the thread inside the block exits the block.