**EX. NO: 9 MULTITHREADING**

**(03/4/2017-9/4/2017)**

**Note: Part I and Part 2 should be recorded in your observation as directed by your mentor**

**Part-I (Who should do Part-I?)**

***Anyone who wants to clear java, hope everyone wants to…….***

1. Write a Java program that executes three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds. Create the three threads by extending the Thread class.

2. Develop a standalone application in Java to demonstrate the concept of thread communication: notify(), wait() methods.

3. Demonstrate the concept of Threads to achieve multitasking program using Thread class and Runnable Interface (separately) for the below scenario:

Create an Array of 9 numbers. And create three Threads to split the task evenly among the three threads. And each thread has to add up and report the answer to the main thread where the main thread waits for the 3 threads and computes the summation of all the three threads. Note: Assign names to the threads as well.

4. Write a program to demonstrate synchronization for the below scenario: There are 2 Telephone booths which has one telephone for service and there is queue of 5 people who can be guided to either of the booths but the condition to be followed is “ A ongoing call conversation of a person should not be stopped in between”.

Part-II: MCQ (Who should do Part-II?)

*Who will not love fun? If you love fun go ahead …and you can take this fun back home and continue there as well……………….*

1. What is the name of the method used to start a thread execution?

A. init(); B. start();

C. run(); D. resume();

2. Which two are valid constructors for Thread?

Thread(Runnable r, String name)

Thread()

Thread(int priority)

Thread(Runnable r, ThreadGroup g)

Thread(Runnable r, int priority)

A. 1 and 3 B. 2 and 4

C. 1 and 2 D. 2 and 5

3. Which three are methods of the Object class?

1. notify();

2. notifyAll();

3. isInterrupted();

4. synchronized();

5. interrupt();

6. wait(long msecs);

7. sleep(long msecs);

8. yield();

A. 1, 2, 4 B. 2, 4, 5

C. 1, 2, 6 D. 2, 3, 4

4. class X implements Runnable

{

public static void main(String args[])

{

/\* Missing code? \*/

}

public void run() {}

}

Which of the following line of code is suitable to start a thread ?

A. Thread t = new Thread(X);

B. Thread t = new Thread(X); t.start();

C. X run = new X(); Thread t = new Thread(run); t.start();

D. Thread t = new Thread(); x.run();

5. Which cannot directly cause a thread to stop executing?

A. Calling the SetPriority() method on a Thread object.

B. Calling the wait() method on an object.

C. Calling notify() method on an object.

D. Calling read() method on an InputStream object.

**Part-III (Who should do Part-III?)**

***If you are a person who loves to challenge yourself, train yourself till you tire and in short for those who aspire to become extra intellect, this is for you***

1. Write a simple Timer that can periodically print a timeout message.

2. Write a simulation program for the fruit market. The farmer will be able to produce different types of fruits (apple, orange, grape, and watermelon), and put them in the market to sell. The market has limited capacity and farmers have to stand in a queue if the capacity is exceeded to sell their fruits. Consumers can come to the market any time and purchase their desired fruits; and if the fruits they want to buy runs out, they are willing to wait until the supply of that kind is ready. (Hint: implementing this market will encounter the producer and consumer problem, and it probably needs multiple buffers for different kinds of fruits).

3. One day a rabbit was boasting about how fast he could run. He was laughing at the turtle for being so slow. Much to the rabbit’s surprise, the turtle challenged him to a race. The rabbit thought this was a good joke and accepted the challenge. The fox was to be the umpire of the race. As the race began (1000m), the rabbit raced way ahead of the turtle, just like everyone thought. The rabbit got to the halfway point and could not see the turtle anywhere. He was hot and tired and decided to stop and take a short nap. Even if the turtle passed him, he would be able to race to the finish line ahead of him. All this time the turtle kept walking step by step by step. He never quit no matter how hot or tired he got. He just kept going. However, the rabbit slept longer than he had thought and woke up. He could not see the turtle anywhere! He went at full-speed to the finish line but found the turtle there waiting for him.

To do Question:

Create two threads, Thread1 (rabbit) & Thread2 (turtle).

Make rabbit to sleep at 750m and sleep for 500msec

Print the winner.

4. You have multithreaded app which looks properly synchronized, but sometimes it seems that when particular field is read it returns old value. You suspect some synchronization issue, how would you investigate?

5. Difference between yield() and sleep()?