

SFDV3006 Concurrent Programming – Lab 1

MCQs

1. Applications that have a GUI should be written as _____
 - a) Sequential Programs
 - b) Concurrent Programs
 - c) Distributed programs
 - d) both a) and b)
2. What is affect of maximising concurrency?
 - a) There is performance gain
 - b) There is performance loss
 - c) CPUs are idle for most of the time
 - d) Depends of the numbers of hardware resources
3. Which of the following uses parallel processing?
 - a) Sequential Programs
 - b) Procedural Programs
 - c) Concurrent Programs
 - d) both b) and c)
4. Which of the following is true about a thread?
 - a) A thread is a heavyweight process
 - b) A thread is a stack in the process
 - c) A thread has data and code
 - d) A thread is a lightweight process
5. Which of the following is the correct way to create a thread in Java?
 - a) Implement Runnable and override the start() method
 - b) Implement the Thread interface and override the run() method
 - c) Overload the run method in the Thread class
 - d) Implement the Runnable interface and override the run() method
6. If *MyThread* implements the *Runnable* interface, what is the correct way to create an instance of it as thread?
 - a) Thread t = new MyThread();
 - b) MyThread t = new MyThread();
 - c) MyThread t = new Thread();
 - d) Thread t = new Thread(new MyThread());
7. To start a thread we should
 - a) first call run() and then start()
 - b) first call start and then run()
 - c) call run() but not start
 - d) call start() but not run()
8. A thread in non-runnable state will transition to
 - a) Runnable state
 - b) Running state
 - c) First Running then Runnable state
 - d) Alive state
9. When the run() method a thread completes it goes to _____ state
 - a) yield
 - b) nonRunnable
 - c) terminated
 - d) sleep

10. When a running thread writes to the disk it goes to _____ state

- a) yield
- b) nonRunnable
- c) terminated
- d) sleep

11. If the *isAlive()* methods on a thread returns true the thread is not in which state?

- a) running
- b) non-runnable
- c) terminated
- d) runnable

Lab Exercises: Creating and using Java Threads

1. Write a class to create a thread *OddThread* that will print odd numbers from 0-100
Write a program called *OddTest* to create and run the *OddThread* class.

2. Write a class to create a thread *EvenThread* that will print even numbers from 0-100
Write a program called *EvenTest* to create and run the *EvenThread* class.

3. Write a program *EvenOddTest* to create one *OddThread* and one *EvenThread* and run them.

4. Assume that you need to calculate a sum of large number of integers which are in a array. The code to sum an array is given below:

```
public int calcSum(int array[]) {
    int sum = 0;
    for(int i = 0; i < array.length ; i++)
        sum = sum + array[i]
    return sum;
}
```

5. Write a class for a thread *PrimeFinder* that will find all the prime numbers between a given range of integers. You must accept the start and end of range in the constructor of the *PrimeFinder* thread.
Write a program *PrimeFinderTest* to create a *PrimeFinder* thread and test that it works correctly

Short Answer Questions:

- 1) What are differences between sequential and concurrent programs?
- 2) Explain some benefits of concurrent programs.
- 3) What is affect of maximising concurrency? Explain your answer.
- 4) What is the *join()* method of a thread and when is it useful?

Long Answer Questions:

- 1) Give an example of at least three different types of programs that are better implemented as concurrent programs.
- 2) What are different states of an alive thread and when it does between those states?