Short Answer:

Answer the following questions with complete sentences in your own words. You are encouraged to conduct your own research online or through other methods before answering the questions. If you research online, please consult multiple sources before you write down your answers. You are expected to be able to explain your answers in detail (Provide examples to each question).

1. What is Thread and What is Process? What are differences between them?
2. How to create threads in Java?
3. Runnable or Thread, which do you prefer to using? Why?
4. What are differences between start() and run()?
5. What if invoking start() method of a thread twice? What if invoking run() method twice?
6. What is Thread Life Cycle in Java? Explain how to get to each stage.
7. Explain join() method in Java Thread.
8. What are differences sleep() and wait()?
9. What is Daemon thread in Java? Why do we need it?
10. What is thread interference? Give an example.
11. What are some of the ways to perform Thread Synchronization?
12. What is Deadlock? How to resolve it?
13. What are the differences between SynchronizedMap and ConcurrentHashMap?
14. What is a Singleton class?

Coding Questions:

Write code in Java to solve following problems. Please write your own answers. You are highly encouraged to present more than one way to answer the questions. Please follow best practice when you write the code so that it would be easily readable, maintainable, and efficient. Clearly state your assumptions if you have any. You may discuss with others on the questions, but please write your own code.

1. Create a program to reproduce the Counter Thread interference issue, run several times and explain the result.

2. Come up with a synchronization mechanism (other than using ***synchronized*** keyword) to make the Counter work as expected.

3. Create a Singleton Class. And write a code to test if your singleton class work in multithread environment?

4. Solve the following deadlock situation:

```
public class driver{

        public static void main(String args[]){

                Object key1 = new Object();
                Object key2 = new Object();

                Thread t8 = new Thread( () -> {
                        synchronized (key1) {
                                System.out.println("t8 has key 1.");
                try {
                    Thread.sleep(5000);
                } catch (InterruptedException e){
                    e.printStackTrace();
                }
                synchronized (key2) {
                    System.out.println(“t8 has key 2”);
                }
            }
        });

        Thread t9 = new Thread( () -> {
            synchronized (key2) {
                System.out.println(“t9 has key 2.”);
                synchronized (key1) {
                    System.out.println(“t9 has key 1”);
                }
            }
        });

        t8.start();
        t9.start();
    }
}
```