Shot 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 is wait(), sleep(), yield()? What are differences among them?
9. What is Daemon thread in Java? Why do we need it?
10. What is thread interference? Give an example.
11. What is memory consistency error? Give an example.
12. What are ways of Thread Synchronization?
13. What is Deadlock? How to resolve it?

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 the main() method to reproduce the Counter Thread interference issue, run several times and explain the result.

2. Use synchronized method to make the Counter work as expected.

3. Write Java Program that implements a multithread application that has three threads.First thread generates random integer for every second and if the value is even,second thread computes the square of number and prints.If the value is odd,the third thread will print the value of cube of number. (hint: Synchronized block)

4. Write a java code for producer consumer problem.
The problem describes two processes, the producer and the consumer, who share a common, fixed-size buffer used as a queue. The producer's job is to generate data, put it into the buffer, and start again. At the same time, the consumer is consuming the data (i.e., removing it from the buffer), one piece at a time. The problem is to make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer. (Do not copy & paste code from online! It will be very helpful for you to understand synchronization in multi-threading env)

- The producer will produce random integers between 1 - 100 and the max size of the queue is 10. Once the Queue is full, producer stop producing random numbers and wait for consumer to consume all the numbers in the queue.

- The consumer will take one integer from the queue at one time, print it with the consumer name, until all the numbers are consumed.
  - Try to implement with only one producer and one consumer
  - Try to implement with one producer and two consumer

5. Considering following code:

```
public class BadThreads {

    static String message;

    private static class CorrectorThread
        extends Thread {

        public void run() {
            try {
                sleep(1000);
            } catch (InterruptedException e) {}

            message = "Mares do eat oats.";
        }
    }

    public static void main(String args[])
        throws InterruptedException {

        (new CorrectorThread()).start();
        message = "Mares do not eat oats.";
        Thread.sleep(2000);

        System.out.println(message);
    }
}
```

The application should print out "Mares do eat oats." Is it guaranteed to always do this? If not, why not? Would it help to change the parameters of the two invocations of Sleep? How would you guarantee that all changes to message will be visible in the main thread?