


PGCert IT: Programming for Industry

Lab 13: Concurrency & Multithreading

Before you start

The BitBucket repository for this lab is located at this link. Fork it to get the starting point for this lab. Take a look at the examples given in the project, as they should help you to complete the exercises.

Exercise One: Simple `Thread` / `Runnable` exercises

1. Write code that declares an anonymous `Runnable` object with the variable name `myRunnable`. The runnable should loop through all integers between 0 and 1 million, and print them out to the console.

```
Runnable myRunnable = new Runnable() {

    @Override
    public void run() {
        for (int i = 0; i <= 1000000; i++) {
            System.out.println(i);
        }
    }
};
```

2. Write code that creates a `Thread` which will run `myRunnable`, then starts that thread.

```
Thread myThread = new Thread(myRunnable);
myThread.start();
```

3. Write code that will request that your thread gracefully terminates, then wait for it to finish.

```
myThread.interrupt();
try {
    myThread.join();
} catch(InterruptedException e) {
    // Probably don't need to handle this.
}
```

Exercise Two: Simple Concurrency

For this exercise, examine the following Java code:

```
public class ExerciseTwo {

	private int value = 0;

	private void start() throws InterruptedException {
		List<Thread> threads = new ArrayList<>();
		for (int i = 1; i <= 100; i++) {
			final int toAdd = i;
			Thread t = new Thread(new Runnable() {
				@Override
				public void run() {
					add(toAdd);
				}
			});
			t.start();
			threads.add(t);
		}

		for (Thread t : threads) {
			t.join();
		}

		System.out.println("value = " + value);
	}

	private void add(int i) {
		value = value + i;
	}

	public static void main(String[] args) throws InterruptedException {
		new ExerciseTwo().start();
	}

}
```

1. What would you expect the output of this program to be?

value = 5050

2. Is the output of the program guaranteed to be what you expect? Why / why not? If not, what is the name of the problem which may occur, and how could you change the program to mitigate this problem?

Not guaranteed, because the add method is being accessed from multiple threads, and is not thread-safe. Certain execution interleavings could cause lost updates.

To fix this I would make the add method thread-safe by adding the synchronized keyword to its signature.