**Java Quiz**

**Read all of the instructions carefully before proceeding.**

1. The quiz is honor system based.

* No communication with any other individual is allowed.
* Internet and book reference is permitted, but the code you write must be your own

2. You have a total of 2 continuous hours from the time you open the quiz to the time you submit it. If you do the bonus, you have 2 hours and 30 minutes.

3. Scan and return all written material including diagrams in a document format.

4. Return all source code (.java files) in one zip file.

5. Sign your full name and total time you took to complete the quiz at the bottom of your file/email.

**Question 1**

Write a class that when executed prints out the numbers from 100 to 1 with the following exceptions:

* If the number is a multiple of 3 print 'foo'.
* If the number if a multiple of 7 print 'bar'.
* If the number is a multiple of both 3 and 7 print 'foobar'.

See attached PrintNumbers.java file

**Question 2**

What is the singleton pattern? Write a class that uses the singleton pattern. If possible, make sure this implementation is thread safe.

It’s a designed pattern to make sure only one instance of the class is created all time. Mainly used for instantiating configuration file in QA Automation.

See attached SingletonExample.java file

**Question 3**

Write a method (you may use more than one) that utilizes the binary search (otherwise known as the half-interval search) algorithm to search a sorted array of Integers (assume the input array has been pre-sorted in ascending order). The return value of the method should be an index of the array that holds the value being searched for, or -1 if the array does not contain the value. Additionally, have the method print out each index that is being examined.

**Example:**

calling binSearch(29, new Integer[] {3,8,22,29,43,55,61,74,78,95,96}) should print 55,22,29 and then return a value of 3.

See attached BinSearch.java

**Bonus Question 4:**

Write a function:

public Iterator<String> combine(Iterator<String> iter1,Iterator<String> iter2)

1. Iter1 and iter2 iterate over their contents in sorted order.
2. The resulting iterator should iterate over the combination of iter1 and iter2 in sorted order.
3. This function should return a Iterator result in O(1) time.

See attached BonusIterator.java