# ICS 211 Fall 2014 Exam 1, October 1, 2014

Clearly write you name on the **front** and **back** of this exam.

This exam is closed-book. No calculators are allowed. There are a total of 100 points. Be sure to answer all parts of each question.

**Question 1** (10 points): Write a static method **even** that takes as a parameter an **int** and returns a **boolean** . If the parameter is >= 0, the value returned should be true if the parameter is even, and false otherwise. If the parameter is < 0, your method should throw the exception **java.io.IOException**, which is *not* a runtime exception.

**Question 2** (10 points):

**public class Hello extends World implements Foo {**

* Does **Hello** inherit all the methods of **World**?
* Does **Hello** inherit all the methods of **Foo**?
* Is **Hello** required to provide all the methods of **World**?
* Is **Hello** required to provide all the methods of **Foo**?
* Is **Hello** a subclass of **World** or **Foo**?

Answer each question (yes or no, and for the last, circle World or Foo).

Explain the reasons for you answers below.

**Question 3** (5 points): Tell me (briefly) everything that is wrong with this method.

**public static double sumProduct(Iterator<Integer> collection) {**

**double result = 0.0;**

**int index = 1;**

**for (Integer x : collection) {**

**result = x.doubleValue() \* index;**

**index = index.next();**

**}**

**return index;**

**return result;**

**}**

**Question 4** (5 points): I have an outer loop from 1 to (1/2)\**n*. The outer loop contains an inner loop from 0 to 3*n*. What is the Big-O of my code?

**Question 5** (5 points): The super class of all Java classes is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_?

**Question 6** (5 points): The Node class for a double-linked list has references to the data and to the next and previous Nodes? **True** of **False.** (circle one)

**Question 7** (10 points):

**static Integer x = new Integer(42);**

**static Integer y = new Integer(42);**

* How many objects are created by the preceding code?
* How many references are in the preceding code?
* Is **x == y** **true** or **false** (circle one)
* Is **x.equals(y) true** or **false** (circle one)

Answer each question and briefly explain your answer.

**Question 8** (20 points): Add the class variable(s) and implement the **hasNext()** and **next()** methods for this Linked List class:

**public class MyLinkedList<E> implements List<E> {**

**protected LinkedNode<E> head; // singly-linked list**

**…**

**private class MyIterator implements Iterator<E> {**

**public Boolean hasNext() {**

**}**

**public E next() { // may throw java.util.NoSuchElementException**

**Question 9** (10 points): I have an abstract data type for a printer. The printer supports two different operations: “printPage”, and “status”. printPage takes as a parameter a string. Status returns a string. Write a Java interface for this ADT.

**Question 10** (5 points): A Java interface is a(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ between the interface designer and the programmer who codes a class that implements the interface. (circle one)

1. precondition
2. postcondition
3. message
4. contract

**Question 11** (5 points): When looping over a circularly linked list, how would you know when you have reached the end?

**Question 12** (10 points): Complete this implementation of the two-argument **add** method for this Array List class:

**public class MyArrayList<E> implements List<E> {**

**protected E [] data;**

**protected int size; // invariant: size <= data.length**

**…**

**public void add(int index, E element) {**

**if ((index < 0) || (index > size))**

**throw new java.lang.IndexOutOfBoundsException(“bad index “ + index);**

**if (size >= data.length) // make room for the new element**

**data = Arrays.copyOf(data, size \* 2 + 1);**

**// your code goes here**