# ICS 211 Spring 2016 Exam 1, February 24, 2016

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

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

**Question 1** (10 points):

**public class Student extends Person implements GradeLevel {**

* Does **Student** inherit all the non-private methods of **Person**?
* Does **Student** inherit all the non-private methods of **GradeLevel**?
* Is **Student** required to provide all the methods of **Person**?
* Is **Student** required to provide all the methods of **GradeLevel**?
* Is **Student** a subclass of **Person** or **GradeLevel**?

Answer each question (yes or no, and for the last, circle Person or GradeLevel).

Explain the reasons for you answers below.

**Question 2** (10 points): Write the **List<E>** interface:

**Question 3** (10 points):

**Integer x = new Integer(314);**

**Integer y = new Integer(314);**

**Integer z = x;**

* 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 4** (10 points): Complete this implementation of the **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 item) {**

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

**throw IndexOutOfBoundsException(“bad index “ + index);**

**}**

**// your code goes here**

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

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

**double result = 0.0;**

**int index = 1;**

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

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

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

**}**

**return index;**

**return result;**

**}**

**Question 6** (5 points): What is the Big-O of the following code?

**double result = 0.0;**

**for (int i = 5; i < 2n; i++) {**

**result = result \* i;**

**for (int j = n – 2; j > 1; j = j - 2) {**

**int temp = j – i;**

**result += temp;**

**}**

**}**

**Question 7** (5 points): The super class of all Java classes is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**Question 8** (5 points): The Node class for a single-linked list has references to the data and to the next and previous Nodes? **True** of **False.** (circle one)

**Question 9** (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 10** (10 points): Implement the following method using the selection sort algorithm:

**public static void selectionSort(E[] data, Comparator<E> c) {**

**Question 11** (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 12** (5 points): When looping over a circularly linked list, how do you know when you have reached the end?