COMP 215 Exam #1

My name is:

- This is a 50 minute exam.
- The exam is open book and open notes. You are free to bring whatever materials you want to the exam. However, the rules are that (a) you cannot use an internet connection during the exam, and (b) you cannot use a Java compiler to help you on the exam (that is, you can't compile or execute any Java code during the exam).
- Write your answers in the spaces provided. Use extra paper to determine your solutions, if needed, then neatly transcribe them onto these sheets.
- Make sure you clearly write your name. Further, please sign the pledge at the bottom of this page.

Pledge: On my honor, I have neither given nor received any unauthorized aid on this exam.

Signed:

Problem 1 [24 Points]

Consider the following code:

```
abstract class TopLevel \{
  abstract void printOne ();
  void printTwo () \{
    System.out.println ("TopLevel");
  \}
\}

class Left extends TopLevel \{
  void printOne () \{
    System.out.println ("Left");
  \}
  void printTwo () \{
    System.out.println ("Left");
  \}
\}

class Right extends TopLevel \{
  void printOne () \{
    printTwo ();
  \}
\}
```

What will the following print out?

```
  ...
  TopLevel one = new Left ();
  TopLevel two = new Right ();
  Left three = new Left ();
  Right four = new Right ();
  one.printOne ();
  one.printTwo ();
  two.printOne ();
  two.printTwo ();
  ...
```

Problem 2 [10 Points]

Imagine that I want to create a generic class `Foo` which takes a type parameter `T`. `Foo` should implement the `List` interface, where the items in the list are of type `T`. Furthermore, `T` must itself implement the `RandomAccess` interface. How would one specify this in Java?

```
class Foo /* tell me what goes here */
{
  /* don't worry about this stuff */
}
```

Problem 3 [30 points]

Consider the following classes:

```
class Foo {
  int value = 0;
  public void increment () {
    value++;
  }
  public void print () {
    System.out.println (value);
  }
}
class RefTester {
  Foo myData = new Foo ();
  Foo getFoo () {
    return myData;
  }
  void setFoo (Foo useMe) {
    myData = useMe;
  }
  void increment () {
    myData.increment ();
  }
  void weirdIncrement (Foo useMe) {
    useMe.increment ();
    useMe = myData;
    useMe.increment ();
  }
}
```

(a) What will the following code print out?

```
Foo temp = new Foo ();
temp.increment ();
RefTester myTester1 = new RefTester ();
myTester1.setFoo (temp);
myTester1.increment ();
RefTester myTester2 = new RefTester ();
myTester2.setFoo (myTester1.getFoo ());
myTester2.increment ();
temp.print ();
myTester1.getFoo ().print ();
myTester2.getFoo ().print ();
```

(b) What will the following code print out?

```
Foo temp = new Foo ();
RefTester myTester1 = new RefTester ();
RefTester myTester2 = new RefTester ();
myTester2.getFoo ().increment ();
myTester1.setFoo (myTester2.getFoo ());
myTester1.weirdIncrement (temp);
temp.print ();
myTester1.getFoo ().print ();
myTester2.getFoo ().print ();
```

Problem 4 [18 Points]

Recall that the `getItem` method for `IDoubleVector` throws an `OutOfBoundsException`. Consider the following code:

```
class ALittleClass {
  double myVal;
  setUp (IDoubleVector dataIn) {
    myVal = dataIn.getItem (0);
  }
}
```

(a) This code will not compile. Suggest one way to get this code to compile by **only** changing the signature for the `setUp` method (that is, without changing any code inside of the method).

(b) Suggest another way to get this to compile by altering the inside of the `setUp` method.

Problem 5 [18 Points]

As you all know, the `ArrayList` class is a generic class that takes a type `T` as an argument.

(a) Why precisely is the following code going to give a compilation warning?

```
...
ArrayList myList = new ArrayList ();
for (int i = 0; i < 10; i++) {
  Integer temp = new Integer (i);
  myList.add (temp);
}
...
```

(b) Beyond the warning, why is this code a bad idea? Be very precise.