1. (2 pts) Suppose we have a class definition called MyString.java which contains a constructor that accepts no arguments. Recalling what the “is a” relationship means (and doesn’t mean), which of the following statements does not generate a compile-time error? (Assume that Object.java does not contain a constructor.) Explain.
   
a) Object o = new MyString(); _____
b) MyString s = new Object(); _____

2. (2 pts) Suppose we have a reference to an int array called numbers. If a Java program contained the following line, would there be a compile-time error reported or a run-time error? Explain. (Assume that the numbers array contains at least one element.)
   
   int average = (numbers[0] + numbers[numbers.length])/2;

3. (2 pts) When an instance variable is declared public or ________________, it can be directly accessed by a subclass, and a list of directories which are searched through by the compiler when we use the import statement is an environment variable called ________________.

4. (2 pts) Suppose we have a class definition called ComplexNumber which contains two instance variables, realPart and imaginaryPart. Suppose there is a constructor and get methods which will return the values of these variables which are called getRealPart() and getImaginaryPart(). Suppose we implement the equals method inherited from Object.java as follows. (Recall that && is logical and.).

   public boolean equals(Object o) {
      return ((o.realPart() == realPart) && (o.imaginaryPart() == imaginaryPart));
   }

   Will the class definition of ComplexNumber compile without an error? If not, what is the problem?

5. (2 pts) Suppose we have a class definition called MyNumber which contains an instance variable of type int called number. Suppose that there is a constructor in MyNumber which accepts one int parameter and initializes the variable number with the parameter. What is printed by the following code segment? Explain.

   MyNumber number1 = new MyNumber(1);
   MyNumber number2 = new MyNumber(1);
   boolean result = (number1 == number2) && (number1 instanceof Object);
   System.out.println(result);