1. What is a Java interface?
   Typically, it is just a collection of method prototypes that are implemented by several related classes.

2. What is polymorphism?
   Sometimes a single variable can refer to objects of different types. For example, if you have an interface called _CanFly_, then you can create a variable of type _CanFly_ which can refer to any kind of object that knows how to fly. (In other words, it can refer to an object of any class which implements the _CanFly_ interface.) This kind of variable is _polymorphic_.

3. Suppose you have an interface called CanDance, and three classes (Student, Penguin, and Cow) all of which implement the CanDance interface. Also assume that there is a method available with the following prototype:
   \[
   \text{public static doSquareDance(CanDance a)}
   \]
   Decide which of the following code fragments are reasonable:
   a. CanDance x = new CanDance();   NO
   b. CanDance y = new Student();   YES
   c. Student z = new CanDance();   NO
   d. Student z = new Penguin();   NO
   e. CanDance a;
      a = new Student();   YES
      a = new Penguin();   YES
      a = new Cow();   YES
   f. Penguin b = new Penguin();
      doSquareDance(b);   YES
   g. Student c = new Student();
      doSquareDance(c);   YES
   h. Cow d = new Cow();
      doSquareDance(d);   YES
   i. CanDance e = new Student();
      doSquareDance(e);   YES

4. ANSWERS TO THIS QUESTION WILL VARY!

5. What is meant by the term _algorithm_?
   A sequence of instructions or steps that can be followed in order to solve a relatively simple well-defined problem. Frequently there are many different known algorithms for solving a problem.
6. Name several problems that can be solved with just an algorithm.
Sorting an array of numbers; Finding the quickest route for going to the store; listing the first 1000 prime numbers; factoring an integer into its prime factors; combining many baskets of eggs into a single basket using a machine that takes two baskets as input and outputs one basket containing all of the eggs in the first two baskets combined; etc.

7. Name several problems that are too complicated to be solved with just a single algorithm.
Running a restaurant; a simulation of a submarine; scheduling the flights for a major airline company; predicting whether or not the economy will improve; etc.

8. What is a _use case_? Imagine that you are working on online banking program. Describe several _use cases_ that your program should be able to deal with. (Recall that there are three parts to the description of a _use case_: the pre-conditions, the actions, and the post-conditions.)

A use case is a typical scenario that a program may encounter; it describes what factors are involved in the case, how the program handles the case, and what you can assume is true AFTER the program handles the case.

For a _banking program_, a typical _use case_ is:

(Withdrawing Money from an ATM)

Pre-conditions: Customer has an account with money in it and has a valid ATM card; ATM has money in it; ATM has paper in order to print receipts.

Actions: Customer puts card into machine; ATM reads card; Customer enters withdrawal transaction; ATM spits out money; ATM spits out receipt; ATM spits out card

Post-conditions: Customer has more money; ATM has less money and less paper