Worksheet Inheritance

The following questions concern the classes on the following page:

1. Which of QA and ChoiceQA is the superclass? The subclass? How is the super-subclass relationship specified? Could a class be both a superclass and a subclass?

2. What methods of QA are overridden in the class ChoiceQA? Why is the method overridden?

3. The keyword “super” is used in the class ChoiceQA. What does it represent and why is it used in both cases?

4. Consider the following code from the Quiz class:

```
QA cqa = new ChoiceQA("What is the capital of California?", "a", choices);
```

What type is the variable cqa? Could it have been declared a different way? Could the variable cqa change type?

5. What if you wanted a list of question-answers, some of the standard type (QA) and some of the subclass type (ChoiceQA). How would you define your ArrayList?

```
public class QA {
  protected String question;
  protected String answer;
  public QA(String q, String a) {
     this.question = q;
     this.answer = a;
  }
  public void displayQuestion() {
     System.out.println(question);
  }
  public boolean checkAnswer (String userAnswer) {
     return this.answer.equals(userAnswer);
  }
  public void displayAnswer() {
     System.out.println("The correct answer is:"+ answer);
  }
}
*****************************************************************************************
public class ChoiceQA extends QA {
  ArrayList<String> choices = new ArrayList<String>();
  public ChoiceQA(String question, String answer, ArrayList<String> choices) {
     super(question,answer);
     this.choices=choices;
  }
```

```
  // override QA's display question so that we include the choices
  @Override
  public void displayQuestion() {
     super.displayQuestion();
     System.out.println(this.choices);
  }
}
**********************************************************************************************
public class Quiz {
     public static void main(String args[]) {
        // create an object of superclass (QA) and use it
        QA qa = new QA("How many states are there?", "50");
        qa.displayQuestion();
        // create an object of subclass and use it.
        ArrayList<String> choices = new ArrayList<String>();
        choices.add("a. Sacramento");
        choices.add("b. San Francisco");
        QA cqa = new ChoiceQA("What is the capital of California?", "a", choices);
        cqa.displayQuestion();
        Scanner scanner = new Scanner(System.in);
        String answer = scanner.next();
        if (cqa.checkAnswer(answer)) {
           System.out.println("correct");
        } else {
           System.out.println("incorrect");
           cqa.displayAnswer() } }
```

6. Explain the difference between the this keyword and the super keyword. When should each be used?

7. Consider the following two automobile classes:

```
public class Car {
   public void m1() {
      System.out.println("car 1");
   }

   public void m2() {
      System.out.println("car 2");
   }

   public String toString() {
      return "vroom";
   }
}
public class Truck extends Car {
   public void m1() {
      System.out.println("truck 1");
   }
}
```

Given the following declared variables, what is the output from the following statements?

```
Car mycar = new Car();
Truck mytruck = new Truck();

System.out.println(mycar);
mycar.m1();
mycar.m2();
System.out.println(mytruck);
mytruck.m1();
mytruck.m2();
```

8. Assume that the following classes have been defined:

```
1  public class Flute extends Blue {
2     public void method2() {
3        System.out.println("flute 2");
4     }
5
6     public String toString() {
7        return "flute";
8     }
9  }
1  public class Blue extends Moo {
2     public void method1() {
3        System.out.println("blue 1");
4     }
5  }
1  public class Shoe extends Flute {
2     public void method1() {
3        System.out.println("shoe 1");
4     }
5  }
1  public class Moo {
2     public void method1() {
3        System.out.println("moo 1");
4     }
5
6     public void method2() {
7        System.out.println("moo 2");
8     }
9
10     public String toString() {
```

```
11        return "moo";
12     }
13  }
```

What is the output produced by the following code fragment?

```
public static void main(String[] args) {
   Moo[] elements = {new Shoe(), new Flute(), new Moo(), new Blue()};
   for (int i = 0; i < elements.length; i++) {
       System.out.println(elements[i]);
       elements[i].method1();
       elements[i].method2();
       System.out.println();
   }
}
```

9. What is the difference between an is-a and a has-a relationship? How do you create a has-a relationship in your code?