Quiz Preparation for Week 6

function and method: interchangable

variable and field: interchangable

instance and object: interchangable

Concepts

- Static function/variable (09/28)
- Inheritance (09/29)
  - overriding functions
- Polymorphism (09/30)
  - "is a" relationships
  - up/down casting
- Equality and Copy Constructor (10/01)
  - overriding equals()
  - designing copy constructor

Warm-ups problems

Static function/variable

1. What is a static function for a class? What's its difference from an instance function?
2. A static function can/cannot do _____?
3. What does keyword *final* do?

Inheritance

4. Bird is a type of animal. A chick is a bird. A penguin is a bird. Declare four classes to reflect their relationships.
5. What is the keyword for invoking parent class constructor?
6. Given the following class declarations, what variable(s) can Bird access from Animal?

```
public class Animal {
  public int one;
  private int two;
  protected int three;
}
public class Bird extends Animal {
  // things
}
```

7. How do you override a function from the parent class?

Polymorphism

8. Identify the error in the following codes:

```
public class Bird { }
```

```
public class Penguin extends Bird { }

void isPenguin(Penguin d) { /* do nothing */}

Bird b = new Bird();
isPenguin(b);
```

9. What is upcasting? Given the following codes, what can Penguin be upcast to?

```
public class Animal { }
public class Bird extends Animal { }
public class Penguin extends Bird { }
```

10. Assume we have the following codes, will there be an error? Why?

```
  public class Penguin extends Bird {
    public void swim() {}
  }

  Bird b = new Penguin();
  b.swim();
```

11. Assume we have the following codes, will there be an error? Why? *(Hint: this is about down casting)*

```
  public class Penguin extends Bird {
    public void swim() { }
  }

  Bird b = new Penguin();
  if (b instanceof Penguin) {
    Penguin thisB = (Penguin) b;
    thisB.swim();
  }
```

Equality and Copy Constructor

12. What are some aspects you want to check for designing your equals() function?

13. How do you define a copy constructor?

Combining it all together!

Whew, that's a lot of materials huh? Let's try to make the rest of material fun and intellectually stimulating!

You are hired by Professor Amazing to do some attendance system design work, for an amazing pay at $4/hour. Illegally underpaid or not, you have no choice but to work, and your next big job is to come up with a structure that manages course information.

1. Design a class named Course, which contains three variables: *name* (String), *attendance* (int), *professor* (String). In addition, you want to have a constructor that will initialize the three variables (make sure to assert that the attendance is non-negative).

2. Although your class Course has quite enough information covered, Professor Amazing wants to specifically add an extension to fit his own course in. He's teaching CS10025 and wants you to, based on Course, add in a new variable to indicate the class *gpa* (double), and print out "This is Amazhingu teaching!" when an instance is constructed. Please fulfill his wishes.

3. Another professor, Professor Splendid is also teaching another section of CS10025. Since she also will use this extended attendance class CS10025, Professor Amazing wants you to override equals() function to make sure the objects can be distinguished. Specifically, (1) the *name* should be the same, (2) if the *professor* is different, then return false.

4. The department is unhappy about Professor Amazing's usage of the non-standard Course extension, as there might be other professors who don't use type CS10025 but still teach CS10025. They want you to make sure the comparison in equals() also takes care of this (the parameter might be of type Course). That is, if the object to compare is type of Course, you should also return true if the course name matches.

Bonus

What if: you want check if a Course/CS10025 object is a given course by comparing to a String? E.g. you have an object other and a String s indicating a course name, and you want to have other.equals(s) to return true if the course name matches.