

CS10, 09S, UCSB

H12: (Based on Horstman, Chapters 9, 10, Interfaces & Inheritance) Total Points: 50 ([printable PDF](#))

Accepted: on paper, in lecture (1-1:50pm) on Monday, May 11th only.
 No email submission allowed.

Name: (1 pts) _____ UCSBNetID (1 pts) _____

Section (2 pts) Circle one: Thu 4pm Fri 10am Fri 11am Fri noon

1. (16 pts) Read Section 9.5 on inner classes, and then answer this question: what are the pros/cons of making `RectangleMeasurer` an inner class, vs. a regular class? Make at least one entry in each box:

	pros	cons
inner class		
regular class		

2. (5 pts) Chapter 11 discusses inheritance. What is the Java keyword used to indicate that one class inherits from another class? To put it another way, how would you fill in the blank:

```
class Student _____ Person
{
...
}
```

3. (5 pts) In the case of `Student` and `Person` (from the previous problem), which one is the superclass?

4. (5 pts) Which typically has more specialized methods and instance variables: superclass or subclass?

Please turn over for more problems

5. (5 pts) Read Section 10.1 through 10.3, and then answer this question.

Suppose you are inside the `toString()` method of a `FacultyMember` class that inherits from a `Person` class. You want to invoke the `Person` class's `toString()` method, and just add an asterisk after what that returns in order to show that the person is a faculty member. Fill in the missing code below.

```
/** toString method for FacultyMember
 * @return what the Person class method's toString returns, plus an asterisk
 */
public String toString()
{
    return          + "*";
}
```

6. (5 pts) Although the textbook uses the term "signature", I wasn't able to find where this term is defined. So, here's the commonly accepted definition: The signature of a method is the stuff that you write before the opening brace, followed by a semicolon. For example, the signature of the `toString()` method given above is:

```
public String toString();
```

Suppose that `t Person` class has a constructor with this signature.

```
public Person (String firstName, String lastName);
```

Here's a constructor for a `Student` class. It is missing one important line—the one that invokes the `Person` constructor. Fill in that line:

```
public Student (String firstName, String lastName, int permNumber)
{

    // now set the instance variables that are specific to student
    this.permNumber = permNumber;
}
```

7. (5 pts) In the previous problem, why is the following an incorrect answer?

```
this.firstName = firstName;
this.lastName = lastName;
```

End of H12