

## Things to Look for on the Classes Test

People who use classes (clients) do not need to know how the methods are written – **encapsulation**

Instance variables should **ALWAYS** be private.

Instance variables are accessible within the class that they are defined.

Inside the class you can use the variable or get methods interchangeably.

Ex in the `Point` class if `x` and `y` are the private instance variables you can access them by using `x`, `y`, `getX()`, or `getY()`

However, you cannot access private instance variables directly from another class like the `Tester` because they are private. The **client program** is any class that has variables of the Class type

```
Student s = new Student("name",100,100,100);
```

```
s.name illegal s.getName() legal
```

If an object has the value `null` you may assign it to another object but you cannot access any of its methods.

```
Point p = null; Point q = p; legal p.getX() illegal
```

Helper methods are methods that are used in the class but are private and are not accessible to a **client program** and must be accessed using an accessor (or get) method. Accessor methods always have a return type that matches the instance variable that they are getting.

```
public double getRate()  
{  
    return rate;  
}
```

Mutator methods are void but have a parameter that the instance variable will change to

<pre>public void setRate(double r) {     rate = r; }</pre>	<pre>public void setRate(double rate) {     this.rate = rate; }</pre> <p>if the parameter name is the same as the instance variables you must use <code>this</code>. to tell the computer you mean the instance variable and not the parameter -- <code>rate = rate;</code> will not give you a compiler error but it will give you wrong results.</p>
--	--

Inside the class the instance variables may be accessed directly or with an accessor method.

**Static methods** in a class are used in a class when a single object has no need to know the information. Example : counting the number of objects created. Any methods that use the static variable must also be static. To access a static variable you must use the class name as the calling object Ex. `Student.getNumStudents`, if `getNumStudents` is a static method in the `Student` class.

When you call methods or constructors the parameters must match in type and number. When you have overloaded methods or constructors (same name but different parameter lists) the computer will use the one that matches you call.

When you call a void method you just use the method name with the parameter list

```
public void discount ( double rate){  
    price = price - price * discount;  
}
```

call     discount(.12); // would discount the price variable in this class by 12%

When you call a nonvoid method you must use the method call in a legal statement so that the returning value has somewhere to go.

```
public double discount ( double price, double rate){  
    return price - price * discount;  
}
```

call     System.out.println("The discount is "+ discount(price, .12));  
or       // would print the price after the 12% discount  
call     double newPrice = discount(price, .20);

example : Assume the Fraction class

```
Fraction f1 = new Fraction (1,2);  
Fraction f2 = new Fraction (1,3);  
Fraction f3 = f2.add(f1);  
System.out.println(f1 + " " + f2+ " " +f3);
```

Trace:

```
f1  1/2  
f2  1/3  
f3  1/2 + 1/3 = 5/6
```

Output:  
1/2    1/3       5/6

You can use print or println with an object if you have defined a toString() method in your class.

Make sure you can write an equals method. You must return true if every instance variable of this object is the same as the instance variables in the parameter object

```
Ex: public class Item{  
    private String name;  
    private double price;  
  
    public boolean equals (Item it){  
        return name.equals(it.name) && price == it.price;  
    }  
}
```

```
or       public boolean equals (Item it){  
        if(name.equals(it.name) && price == it.price)  
            return true;  
        else return false;  
    }  
}
```

You should be able to write a simple class or methods in a class.

Review your 3 worksheets. Understand the Multiple Choice WS. Review your quiz.  
Be familiar with the Car Lab, Name Recognition Lab, and the Fraction Lab

65 points     38 points (2 points each)   multiple choice  
              27 Free Response   (9 each for each AP style question)  
Partial credit is given