CS 2300 Week 06 Lab

A simple class: TwoInts

This week's lab is to develop a simple class. This is very similar to the example class used in the classroom presentation.

1. Start a lab as usual: (Fire up Intellij IDEA. Create a new project named `TwoInts`. Use something like `lab06.TwoInts` for the package and class name.

2. Put a **`main` method** inside the class. The main method should stay at the **end** of the class, so put everything else above it. One easy way to do this is click on the line just before the closing `}`, and type `psvm`, and hit the tab key. The main method will be used to hold tests for the rest of the class.

3. Add to the comment block at the top of the file (above the class name), so that it looks something like this:

```
/**
 *   Your name here
 *   CS 2300
 *   Lab 06:
 *   TwoInts class
 *   The date here
 */
```

4. The purpose of this class is to hold some **data**, namely two `int`s. The variables that hold the data are called **fields**. Fields usually go at the top. These fields are both `final int`s with the names `a` and `b`.

Add the line:

```
    private final int a;
```

for a field to hold `a`. Add a similar line for `b`.

5. After the fields come the **methods**. There is usually one or more methods called **constructors** (or, **c-tors** for short). Any c-tors usually come right after the fields. The c-tors have the same name as the class.

5.1. Add a **full service c-tor** (constructor) to initialize the two fields. Write it like this:

```
public TwoInts (int a, int b){
    this.a = a;
    this.b = b;
}
```

5.2. Add a **one arg c-tor**. This can be used for the event that you want both `a` and `b` to have the same value. Do it like this:

```
public TwoInts (int both){
    this(both, both);
}
```

Here is what is going on: inside a c-tor, you can call **another** c-tor (for the same class) with the statement `this();`. Here we send two copies of the input argument `both` to the c-tor that expects two arguments.

5.3. Add a **zero arg c-tor**. It is usually a good idea to have a zero arg c-tor to act as a default. It should use good default choices for the values. Zero is a good default.

6. After the c-tors, make a **`toString`** method. The `String` returned by the `toString` method should look like this: `<5, -3>` if the values in the fields `a` and `b` are 5 and -3. The **signature** of this method must be this:

```
public String toString(){}
```

Put some stuff inside the `{}` to make it return a string containing the info as specified above.

7. Now try it out. We will use the existing `main` method to test everything. (This should be at the end of the class).

Put the following code inside the main method:

```
TwoInts blue = new TwoInts(4, 11);
```

Now hit `soutv` and hit TAB. Intellij IDEA will add the following line:

```
System.out.println("blue = " + blue);
```

This trick makes an easy check for the most recently defined variable. This is VERY USEFUL!!! Run the program and see what happens.

Now add

```
TwoInts green = new TwoInts(14, -1);
```

For this and each other new object you add, do the `soutv` + TAB trick so you can test it. This is a very easy way to test EVERYTHING as you write it.

Add the following to test the two other c-tors:

```
TwoInts one = new TwoInts(144);
System.out.println("one = " + one);
TwoInts zero = new TwoInts();
System.out.println("zero = " + zero);
```

Fix anything that is not working properly.

8. Right after the `toString` method and before the `main` method, add **getters** for each field. The first one will be

```
public int getA(){
    return a;
}
```

This very simple method will return a copy of the field `a`.

Test this by adding the following to the main method:

```
System.out.println(“blue.b is: “ + blue.getB());
```

Be sure this works as expected.

9. Now add a method named `aIsPos` that returns `true` if `a` is positive, and `false` otherwise. Figure out what the signature should be, and write the method!

Put the following code in the main method to test the new method.

```
if (green.aIsPos()){
    System.out.println("Yep!");
} else {
    System.out.println("Nope!");
}
```

Run this and see if it works.

10. **What to Hand In**.

Upload the file containing your program code to Blackboard in the usual way.