

## Chapter 5 Quiz

1. [2pts] What does SRP stand for?
  
2. [2pts] What does DRY stand for?
  
3. [14pts] Complete the following SRP analysis.

```

Automobile
+ start() :void
+ stop() :void
+ changeTires(tires : Tire[]) :void
+ drive() :void
+ wash() :void
+ checkOil() :void
+ getOil() :int
    
```

**SRP Analysis for** Automobile

  

The \_\_\_\_\_ itself.

The \_\_\_\_\_ itself.

The \_\_\_\_\_ itself.

The \_\_\_\_\_ itself.

The \_\_\_\_\_ itself.

The \_\_\_\_\_ itself.

The \_\_\_\_\_ itself.

<b>Follows SRP</b>	<b>Violates SRP</b>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>

4. [6pts] Match each of the following techniques to what it does.

Unplanned tasks and user stories

I help you make sure that everything has its place, and that place is only **one** place.

Perfect design

With me, the design gets better with small improvements throughout your code.

SRP

I make sure that the unexpected becomes the expected and managed.

Refactoring

My mantra is, "Perfect is great, but I deliver."

DRY

I make sure that all the parts of your software have one well-defined job.

Good-enough design

I'm what you strive for, but ultimately you might not deliver.

5. [1pts] If you find you are cutting and pasting large blocks of your design and code then there's a good chance that you're breaking the \_\_\_\_\_ principle.