

1 – Literary Quotations - creating a class and method invocation

1. Create a class to print to screen some famous quotes.

Quotes

```
+ shakespeareQuote() : void  
+ oscarWildeQuote() : void  
+ churchillQuote() : void  
+ georgeWBushQuote() : void
```

The output of the program when you invoke each method should be...

```
Continuous effort - not strength or intelligence - is the key to unlocking our potential. Churchill  
Good night, good night! Parting is such sweet sorrow, that I shall say good night till it be morrow. Shakespeare  
Experience is simply the name we give our mistakes. Wilde  
One of the great things about books is sometimes there are some fantastic pictures. George W Bush
```

2. Update your program to include the following quote ...

"If the facts don't fit the theory, change the facts. Albert Einstein"

3. Test your program again.

2 – Simpsons

Create the following class to represent a Simpsons character in an application.

SimpsonsCharacter

```
- name : String  
- catchPhrase : String  
+ getName() : String  
+ setName(String) : void  
+ getCatchPhrase() : String  
+ setCatchPhrase(String) : void  
+ sayCatchPhrase() : void
```



Create another class **CartoonApp.java** to create each following characters. (It should have a main method).

| Character name | Catch phase |
|----------------|--------------------|
| Bart | Eat my shorts! |
| Homer | D'Oh! |
| Lisa | I'll be in my room |
| Nelson | HaHa |

For each of the character objects set the appropriate instance variables, e.g. for the *Bart* character set the Name to "*Bart*" and the Catch Phase to "*Eat my shorts*". Then for each character object in turn invoke the **sayCatchPhrase** method. The **sayCatchPhrase** method should output to screen ...

<CharacterName> says <CatchPhrase>

So the output to screen should show

```
Bart says eat my shorts !  
Homer says D'Oh !  
Lisa says I'll be in my room  
Nelson says HaHa
```

Part 3 – Calculator - creating classes – invoking methods with parameters

1. Create a class to represent the operations of a basic calculator.

Calculator

```
-memory : int  
+addNumbers(int, int) : int  
+ subtract(int, int) : int  
+multiply(int, int) : int  
+divide (int,int):int  
+sqrRoot(int) : double  
+setMemory(int) : void  
+getMemory() : int  
+clearMemory() : void
```

Create an instance of the calculator and test all the functions. Typically a class in Java is tested by the author using an automated test program (e.g. *JUnit*). We will be doing this later in the course but test using test specification like this ...

| Test Case | Test description | Test data | Expected outcome | Test results (screen capture or observed outcome) |
|-----------|--|----------------------|---------------------------------|---|
| 1.1 | Testing the addNumbers method. Two numbers will be passed to the method and the answer checked against the expected answered. | Num1 = 3 Num2 = 4 | Answer should be displayed as 7 | |
| 1.2 | Etc | | | |
| | | | | |

....

Part 4. Car factory - creating classes.

OO Examples

1. Create a Java class to represent a **Car**. It should have the following properties (state) and methods (behaviour).

State

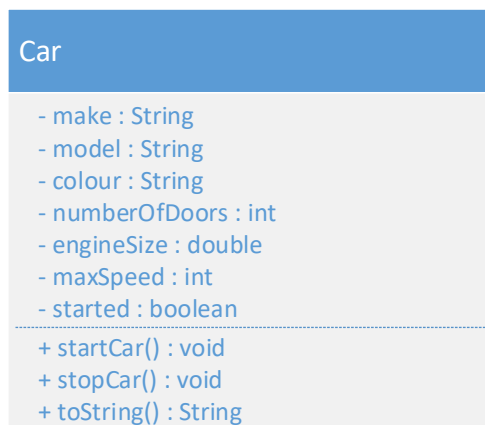
Make
Model
Colour
Number of doors
Engine size
Max speed

Started

Methods

It should be possible to start the car (i.e. create a method that confirms the car is started – output to screen) and stop the car (also a method that outputs that the car is stopped). These methods should be able to read and alter the *started* property

Represented in a UML class diagram...



The **toString()** method should return all the car properties.

An example of the **stopCar()** & **startCar()** methods... but do we really need to expose the **isStarted()** and **setStarted()** methods...? Should we check the value of the started when we attempt to start or stop the car ? Trying to start a car that is already started may well damage the starter motor...

2. Create each car with the specification details as detailed above and output the properties to screen.

| Make | Model | Colour | Doors | Engine Size | Max speed |
|---------|--------|--------|-------|-------------|-----------|
| Ford | Fiesta | Blue | 4 | 1.2 | 110 |
| Peugeot | 308 | Silver | 4 | 1.8 | 130 |
| Ferrari | F4 | Red | 2 | 2.8 | 230 |