Assignment 10

Composition

The point of this assignment is to cover the primary concepts of composition.

Using the IDE

Remember that all code must be submitted in text format.

Initial Code & Output

Load the following code into the online compiler:

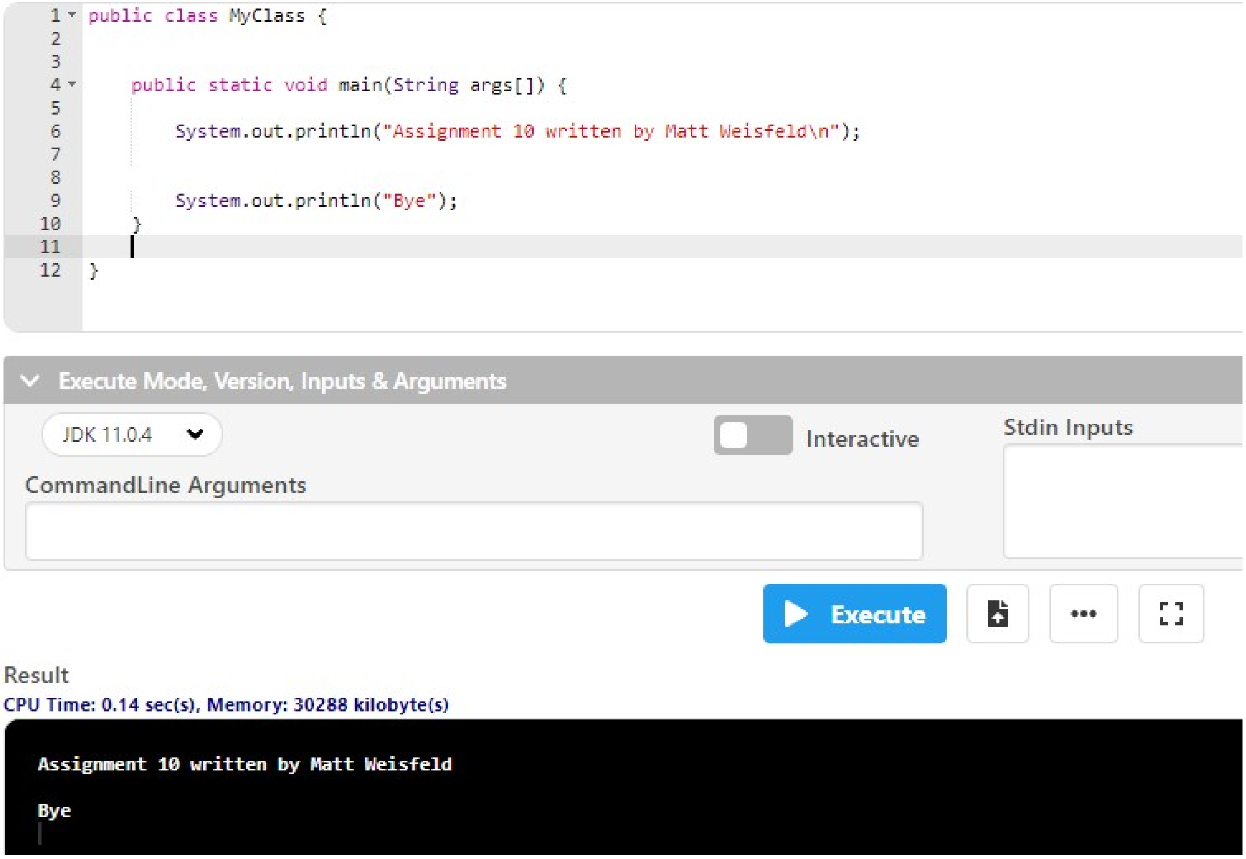
https://www.jdoodle.com/online-java-compiler/

Please copy the following code into the IDE, compile and run it.

```
public class MyClass {

   public static void main(String args[]) {

      System.out.println("Assignment 10 written by Matt Weisfeld\n");

      System.out.println("Bye");
   }

}
```

This code you will be writing covers a lot of programming concepts. Please take this opportunity to study the code and determine what is going on at each step.

When you execute the code it will look something like this:



```java
public class MyClass {


    public static void main(String args[]) {

        System.out.println("Assignment 10 written by Matt Weisfeld\n");


        System.out.println("Bye");
    }

}
```

JDK 11.0.4

Result

CPU Time: 0.14 sec(s), Memory: 30288 kilobyte(s)

```
Assignment 10 written by Matt Weisfeld

Bye
```

Not much is going on here – yet ☺!

Problem

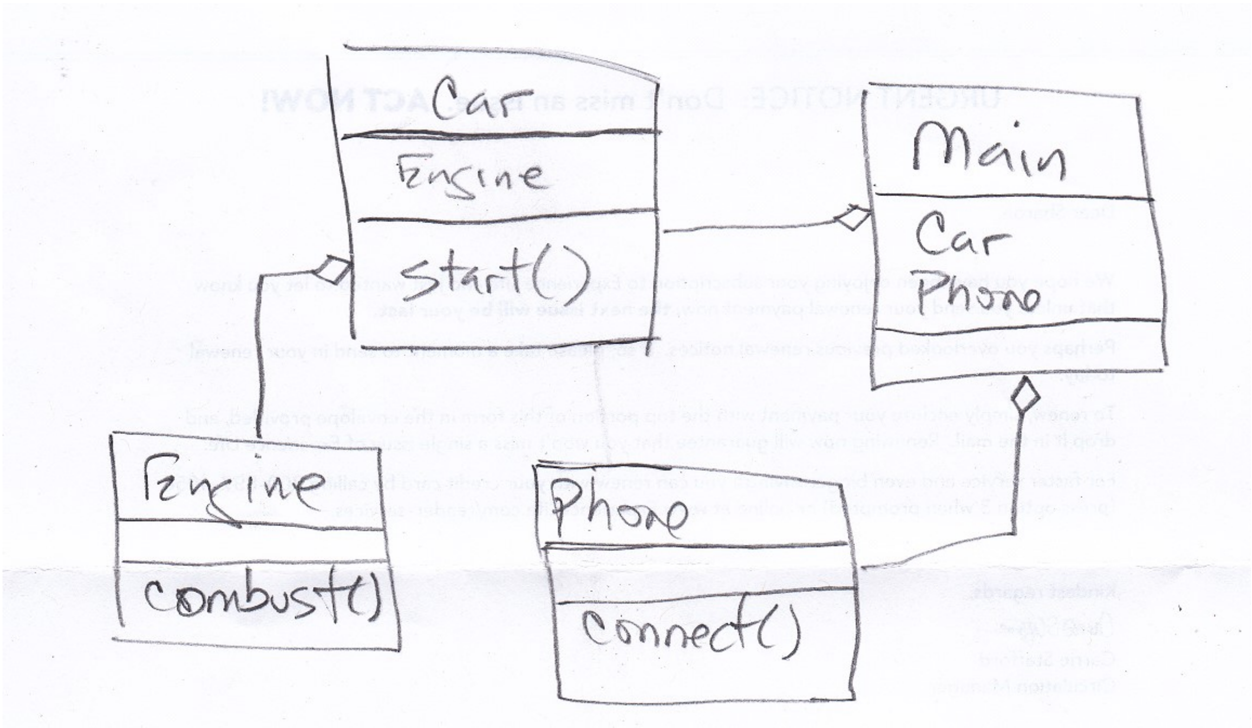
Note that this is simply a shell to get you started with a clean compile.

The task is to create a Car class that follows the following UML specification. Then the main() application must create a car (named mustang) and a phone (named android) .

The scenario here from the Car's perspective is to create an aggregation relationship with Engine and an association relationship with the phone.

The idea is that the car:

1) Must have a single engine that cannot be separated from the car (*at least not by the driver* ☺). Thus, the Car and the Engine combine to form an **aggregate**.
2) Can connect to a phone via Bluetooth when the car starts and thus can be connected to any phone that has Bluetooth – forming an **association**.



Here are the constraints that you must include in your program.

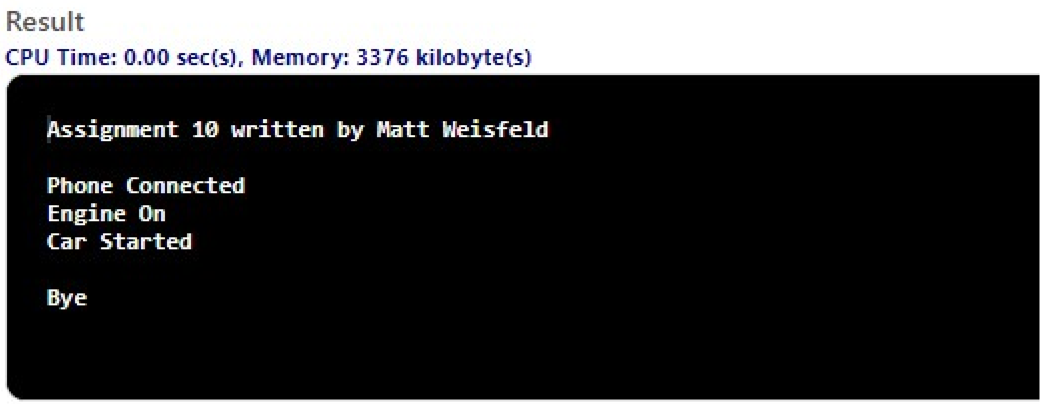
1) Include an output statement at the beginning of the program with the assignment number and your name:

   A10 – Written by Matt Weisfeld

2) Create a **single** Car object called *mustang* within the *main*() application.
3) Create a **single** Phone object called *android* within the *main*() application.
4) The Car class has an attribute representing the aggregate Engine.
5) The Car class has a single method called *start*().
6) Create an Engine class with a single method called *combust*().
7) Create an Phone class with a single method called *connect*().
8) When the Car's start method is invoked, invoke the *connect*() method of the phone.
9) When the Car's start method is invoked, invoke the *start*() method of the engine.
10) After the main method creates the mustang, invoke the mustangs *start*() ,method.
11) The phone should print out that it has connected.
12) The engine should print out that the engine is on.
13) Print out an exit greeting (Bye) when exiting the application.

Final Output

Once completed, your output (in the following test case) should look like this:



Result

CPU Time: 0.00 sec(s), Memory: 3376 kilobyte(s)

```
Assignment 10 written by Matt Weisfeld

Phone Connected
Engine On
Car Started

Bye
```

- Note the input box

What to Submit

A single Java text file should be submitted to Blackboard.