1. Consider the following class:

```
class Car {
    String type;
    int year;
    boolean electric;
}
```

After the following line of code is executed, what are the values of `c`'s member variables?

```
Car c = new Car();
```

`c.type`	?	`null`
`c.year`	?	`0`
`c.electric`	?	`false`

2. Fill in the constructor for the following class. (The paremeter `type` should be assigned to the member variable `type`, the same should be done with `year` and `electric` – nothing more is needed).

```
class Car {
    Car(String type, int year, boolean electric) {
       // code goes below



        this.type = type;
        this.year = year;
        this.electric = electric;



    }

    String type;
    int year;
    boolean electric;
}
```

3. Using only the code from question (2), create an object of type `Car` called `c`. (Give its member variables whatever values you would like).

```
Car c = new Car("Ford", 1991, false);
```

4. What is wrong with the following code? Explain below.

```
class Car {
    String type;
    int year;
    boolean electric;

    public static void main(String [] args) {
        int year = 1998;
        this.year = year;
    }
}
```

`this` refers to the implicit object. You cannot use the `this` keyword in a `static` context, as there is no implicit object. It may only be used from within a member function.

5. What is wrong with the following code? Explain below.

```
class Car {
    void setType(String type) {
        this.type = type;
    }

    String type;
    int year;
    boolean electric;

    public static void main(String [] args) {
        setType("Ford");
    }
}
```

`setType()` is a member function of the `Car` class. Member functions must be called through an object – i.e. `carObj.setType("Ford")`