Homework II

Policies

- This homework is due Monday, 10/20/2014, 12:05 am.
- Late hand-ins will be accepted until the same day, 08:00 am. They will be penalized with 10% of the maximum achievable score.
- If you have any questions, you can use the Piazza forum for our class: https://piazza.com/uci/fall2014/141/home - please make sure to select the correct folder. Do not use Piazza to post any partial solutions.
- Include your name and your student id on all pages and in all source code files.
- The submission process is described below.

Question 1 (3 Points)

Briefly describe the concept of virtual method tables (vtables). Why are they needed and how do they work?

Question 2 (1 Points)

Is there a disadvantage in using virtual method calls? If so, what is it?

Question 3 (3 Points)

In C++ and similar languages, what are the differences between non-virtual function, virtual functions, and purely virtual functions?

Question 4 (3 Points)

table.c (in EEE Dropbox) gives you the frame for a very simple calculator for integers. You call it with ./calculator OP A B where OP is the operation code, encoded as 0 for addition, 1 for subtraction, 2 for multiplication, and 3 for division. A and B are the two operands.

The main method performs an operation by looking up the operation code in an array of functions and then calling the appropriate function.

Implement the calculation functions and add the array of operations. If you need help, search the internet for function pointers for C and how to store them in an array.

Question 5 (10 Points)

The class diagram on the next page shows a hierarchy that we implemented in Java[1]. You will find the source code for this in the Dropbox.

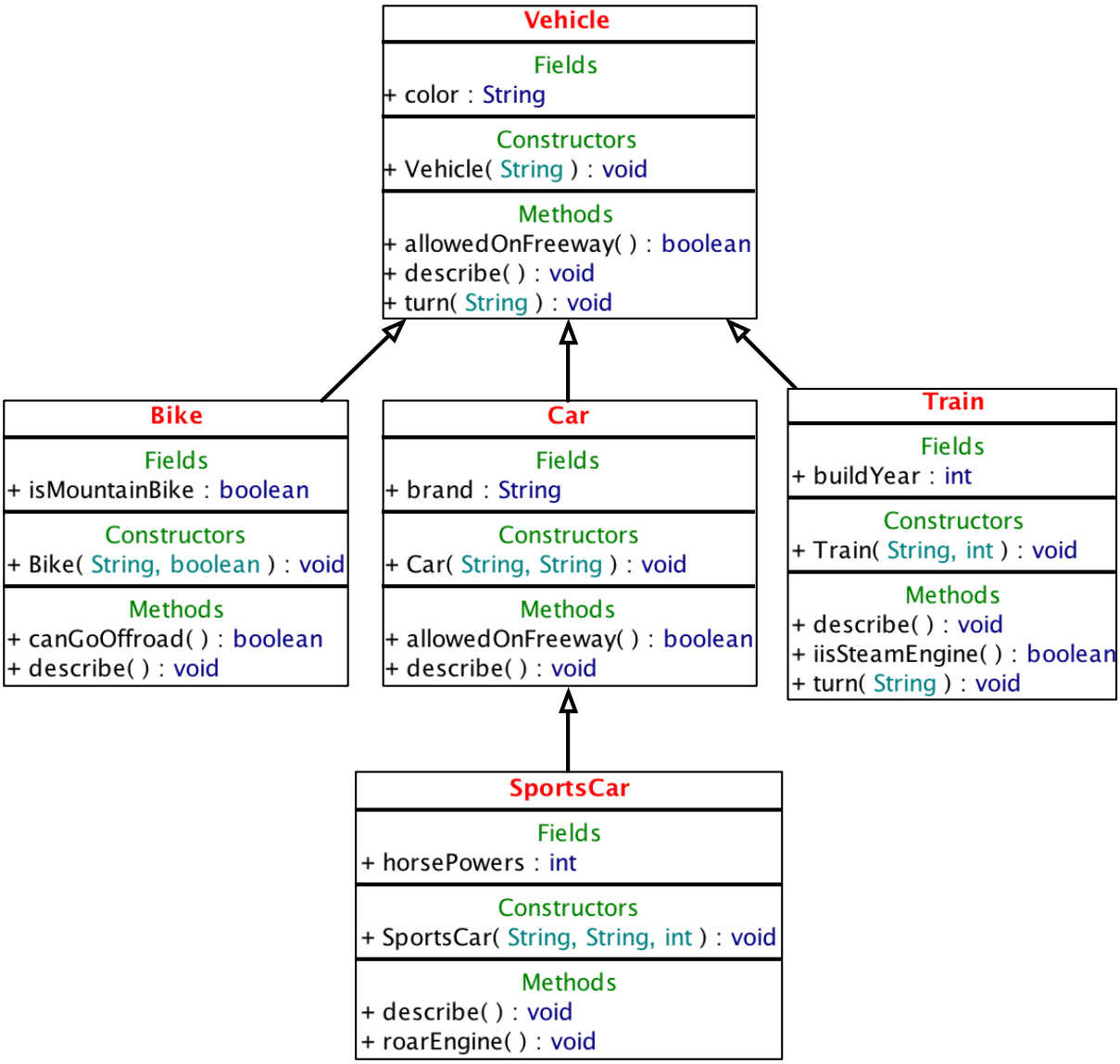
1. Vehicles
    a. All vehicles have a color. They are constructed with this color as a parameter.
    b. Vehicles can be asked if they are allowed on the freeway. If this is not overridden by a subclass, the default is to not allow them there.
    c. Vehicles can be asked to describe themselves. All subclasses override this method to give details about themselves.
    d. Vehicles can turn. The direction is given as a string ("right" or "left").
2. Bikes
    a. Bikes are vehicles. In addition to the known parameters they can be mountain bikes, as defined during construction.
    b. Bikes can be asked if they can go offroad. Mountain bikes can, others cannot.
3. Cars
    a. Cars have a certain brand. This is passed as part of the construction.
    b. Cars are allowed on the freeway. Thus, they need to override the allowedOnFreeway() method
4. SportsCars
    a. Sports cars are cars. For them, we additionally store the horsepowers.
    b. Sports cars can roar their engines.

Your job is to simulate this class hierarchy in C. Since C is not an object-oriented programming language, you will need to simulate concepts like inheritance and polymorphism (overriding methods). This is similar to what first versions of the C++ compiler did (then called C with Classes).

First steps were already done in VehiclesTest.c (also available in the Dropbox). Vehicle and Bike are already implemented. Your job will be to implement the remaining three classes.

To compile this using gcc, you will need to add the -fms-extensions flag to the command line of your compiler. This has been tested on the Linux lab machines (which you can also access from home via VPN at openlab.ics.uci.edu). For Virtual Studio under Windows, you might not need this additional flag.

[1] Created using class-visualizer.net



Submission

Please submit your answers to the EEE Dropbox for this homework (e.g. COMPSCI 141 HW 1). You can find instructions on how to do this here: https://eee.uci.edu/help/dropbox/students/#studentsubmit

Submit a zip file with the following contents:

- **answers.pdf**: Exercises 1 thru 3 in the PDF format, include your name and your student id on all pages.
- **table.c:** Your solution for exercise 4, include your name and your student id as a comment
- **VehicleTest.c:** Exercise 5, include your name and your student id as a comment

In addition to reading your code, we will submit it to an automated test suite. For this reason, it is important that your submission follows this format.