Designing a GPS System
5004 Object Oriented Design

In Module 7, we started exploring broad design patterns that are less about syntax and more about the scope of an entire program. In this lab, you will be given specifications for what a program will have to model. You will also have to design a view system for displaying that information to a user. Finally, you will have to build a control unit to manage these two systems.

1. Goals:

- Build a working program that uses an MVC design pattern
- Deepen your understanding of program design
- Implement technical solutions that draw from what we have learned so far
- Follow SOLID principles appropriately

2. Instructions:

This assignment asks you to design some basic calculations for a simplified GPS system. Do not worry about making your program behave exactly like a real-life GPS. We don't have the time or tools to do that. Instead, design a Model View Control style program that provides the following functionality:

1. Calculate the distance between two points on a map, given their latitude and longitude.
2. Calculate the time it will take to arrive at a destination given an average speed and the distance to travel.
3. Calculate the amount of time remaining until a vehicle will have to refuel given a miles-per-gallon consumption rate and a current speed.
4. Display all this information to the user through an output of some kind.
5. Design a driver file that easily takes in a baseline of information (for example, a vehicle's fuel efficiency and gas in the tank. However, the driver should also be able to be updated with new commands that adjust calculations.
6. Ensure that your program adheres to SOLID principles of design.

Bonus Challenge (not required):

- Calculate the time needed to bring a car to a complete stop given:
    - A vehicle's current speed
    - Distance from the desired stopping point (e.g. a red light or a stop sign)
    - An assumed constant deceleration rate

Expected Files:

You can turn in a program that uses more files than the ones listed below (for example, maybe you want to leverage an enum or interface), but at the minimum, you should design a program that has

- A driver file
- A model file
- A view file
- A controller file

3. Reflection:

Each assignment must include a short reflection. The generation of it should take you no more than 15 minutes. This gives you a chance to reflect back on what you learned. For this reflection, please answer the following questions:

1. What did you learn that you were not expecting to in doing this project?
2. How did the required MVC design change your approach to building a program? Did it help or did it make things more difficult?
3. Do you have any questions about MVC?

4. Submission:

Please read carefully. Failure to follow submission instructions can result in a reduced score.

Submit all files on Canvas under the appropriate assignment. Make sure to include the following:

Submit your files as a single zip file named: `“Your Name_Assignment”.zip`

Your zip file should contain all files needed to make the application run. Include your reflection as: `“lab_8_reflection.pdf”`

Submission checklist:

- □ Did you include adequate comments?
- □ Did you include comment blocks at the top of each file?
- □ Did you name your files as requested?
- □ Does your code compile?
- □ Did you take care of any warnings presented by your IDE?