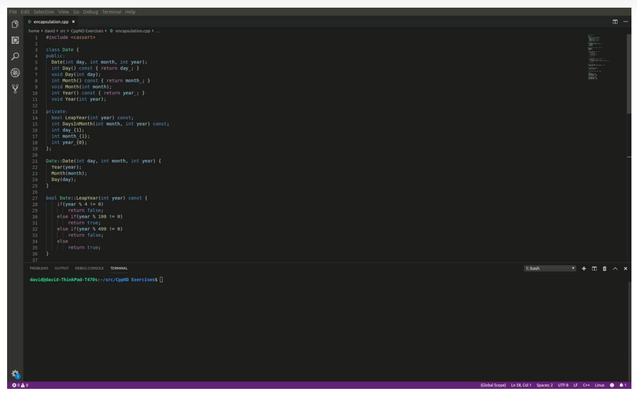

https://youtu.be/-WRtCE3MZB8

Mutator Functions

A mutator ("setter") function can apply logic ("invariants") when updating member data.

Exercise: Car Class

In this lab you will create a setter method that receives data as an argument an converts it to a different type. Specifically, you will receive a string as input and convert it to a character array.

1. Create a class called `Car`.
2. Create 3 member variables: `horsepower`, `weight` and `brand`. The `brand` attribute must be a character array.
3. Create accessor and mutator functions for all member data. The mutator function for `brand` must accept a C++ string as a parameter and convert that string into a **C-style string** (a character array ending in null character) to set the value of `brand`.
4. The accessor function for the `brand` must return a string, so in this function you first will need to convert `brand` to `std::string`, and then return it.