

## C Programming: Object Orientated C Tutorial

### Part (a): Creating a branch to work on

From the master branch of the `empty-c-project`, create a new branch for you to work on; useful will be the `git` commands `branch` and `checkout`.

### Part (b): Creating a simple API

We want to create a simple API using two new files, a `.c` and a `.h` file.

Create these two new files in the appropriate locations and rerun `cmake` to ensure your `Makefile` is aware of the latest files.

Populate your files to achieve the following:

- A private struct that contains an `int`, `double` and a `string`. You might need to give the `string` a bit of thought; it is common to see `string` lengths limited by some global `MAX`.
- Write set functions for all three members, ensuring that the integer is `>0`, the double cannot be `<2`, and the string has to start with the letter 'H'.
- Write two get functions, one that returns a `double` that is the integer multiplied by the double, and the second returns a **copy** of the string.
- Write a function that prints a summary of your structure.
- In your `main.c`, write code that tests that all of these things work.

### Part (c): Adding the ability to spawn multiple objects

We want to expand our API to allow us to create and use multiple objects referenced by a typecasted void pointer handle, i.e. `typedef void *my_struct_handle`. To enable our solution to the first part of this tutorial to stay unmodified, we want to commit our solution to our current `git` branch, then create and change it to a new branch for our solution to the next part.

Once you have committed to your solution and switched branches, implement the following:

- Create a function that initialises an instance of your structure, returning a handle to the structure.
- Modify all of your existing functions to take in a handle to a structure as their first argument.
- Update your existing functions to typecast the provided handles to use the provided structures.
- Update your main code to test that all your existing functions work.
- Write a function that takes in two structs and returns the value of the first structure's integer multiplied by the second structure's double.
- Add a test to your main that tests this last functionality.
- Commit this solution to your new `git` branch.
- Using `git merge`, merge this solution to the master branch of the `git` repository.