

## Programming Assignment #1

### CS 202 Programming Systems

**\*\*\* Make sure to read the Background Information first!**  
**It applies to all programming assignments this term\*\*\***

*We do not accept late work beyond the late due date.*

*Always backup your files PRIOR to using tar*  
*Double check that the arguments specified with for tar are correct!*  
*Late penalties will apply to all submissions incorrectly archived/uploaded*

#### **Background:**

When beginning with this project, the first thing to keep in mind is that we are no longer working on CS163 programs! In CS163 we were concerned about creating Abstract Data Types and the class construct facilitated this. Instead, this term we will be focusing on how to create Object Oriented Solutions. An ADT may be part of that solution – but it certainly shouldn't be the primary focus. Instead you want to strive for classes to have specific “jobs” and have classes derived from more general classes, whenever appropriate. We will be working in situations where there are multiple classes, so you will want to focus on dividing the design into smaller components that have specific jobs working together to solve the problem.

Every assignment this term needs to have at least 5 classes. With these, think about how to design the classes such that they reduce the amount of work another class needs to do. The idea is if we have “robot” like classes doing the smaller tasks or “jobs”, that by the time we get to a larger class that has more to manage – it will have little left to do! We can achieve this by delegating. Often the over-use of “getters” can cause the opposite to happen – and instead of delegating the managing class has to fundamentally do all of the work itself.

#### **Overview:**

Drones have become very popular. They are now used for wide variety of activities from military to entertainment. Not only are there different purposes, there are also different kinds of drones. Here is just one list I discovered when doing my research:

- Quadcopters and Helicopters
- Racing, Selfie, Toy, Professional Drones
- Devices that support a GPS system
- Drones that are ready-to-fly
- Devices created especially for Photography
- Delivery drones and endurance drones
- Devices that support a different fueling system (gas-powered, nitro-powered)

Drones may be used for recreational purposes. These usually will have smaller motors that can be charged using batteries. Some smaller drones may only be used for indoor purposes. Others need to fly outdoors. Aerial Photography is also a common use by drones. My daughter's wedding will be shot using a drone to get footage from across the water that would be impossible otherwise. These can take close-ups and therefore require stabilizing features. Then there are drones that are used for surveillance and security. Since these may need to go a farther distance, they will need increased stability and extended battery life. Therefore, drones have size, battery life, size of motor, and at least stabilizing features to consider.

Drones also have many different kinds of actions. You can take off and land. Drones can move in three dimensions: yaw, pitch (e.g., up and down), roll (e.g., rotation front to back). Drones should never collide with one another nor go above the approved ceiling for flying in a particular area. They also cannot fly within 5 miles of an airport or near emergency response efforts.

### **Program #1**

For Program #1, you will be creating an object oriented program that will simulate at least three of the kinds of drones. Read the above and then research the type of drones that you are interested in. Once you make a decision about the type of drone you want to support – select at least two different types of uses for each of your three drone types.

Once this has been decided, the next step will be to think about breaking this down into a series of classes and create them independent of the entire problem. Some relationships should be hierarchical, others can be containment. With hierarchies always push the common elements up to the base class. And Nodes will need to be classes instead of structs.

Then once you have decided upon the desired classes and relationships, then start mapping out operations that will control the drones. Avoid classes with only setters and getters! It is during this step you need to consider how two different drones will ensure that they do not collide! Support should be in place such that a drone will not let itself enter a restricted area.

The following represents some ideas on the design – the first step is to plan what classes might make the most sense! Some of the basics that are part of an OO program **could** be:

1. Controller class – which controls the movement of a drone
2. General drone base class
3. Three specific types of drones derived from that base class

Anything that is similar between these or other classes that you write should be pushed up to be part of a base class. For example, classes that manage collections of items may be derived from a common base class that manages the collection. Keep classes small and functions small. A large class or function means that the problem has not yet been broken down into its basic components (objects).

### Data structures

This program should implement a Graph to handle the location, including elevation, of the drones and restricted areas.

- a. Your graph should be implemented using an adjacency list which is a dynamically allocated array of head pointers to edge lists. Each head pointer points to a LLL of adjacent drone locations.
- b. All traversal functions should be implemented **recursively**.