

# Extra Research Points

## Friend functions

Friend functions are functions which are not implemented as a class definition, but still nonetheless has access to the private and protected variables in a class. The function `getSecrets()` in my code is an example of a friend function. This function has direct access to the private variables such as first name, last name, address etc. instead of using the getter functions for each variable.

## Virtual Functions and Abstract Classes

Abstract Classes are Classes that implement virtual functions. Pure abstract classes are classes that are never instantiated, only inherited from. In my code, there are not pure abstract classes, as I use the Person class to demonstrate upcasting and downcasting classes.

Virtual Functions are declarations for functions that force children to uniquely implement the function. In my code, the virtual function `toString()` in the Person class forces Student and Teacher classes to both have a `toString()` function, both of which have different functionality.

## Upcasting/Downcasting classes

Upcasting and Downcasting classes is effectively converting an instance of a class between different levels of inheritance. In my code, I upcast each instance of student and teacher to the Person class by placing it in an array which contains Person objects. Then I iterate through each object and call functions that are shared between objects.

## Rule of 3

The rule of 3 in C++ programming refers to 3 overloaded functions of copy constructor, assignment operator, and destructor. These 3 functions are often written to implement additional logic in addition to that of C++ standard functions for these operations. In my code, I wrote these three functions for all of my classes.