**School Directory Classes**

By: Kyle Riebeling

This program is meant to test the capabilities of the Person, Student, Athlete, Teacher, and Experience classes. When the program is run, a menu is displayed along with a greeting. You will select a menu option for the type of person you are looking to enter the data for. After selecting the option, you will be prompted to enter all needed information for that object type. As you enter the data, it will be saved to the object of that class type. If improper input is entered, the program will tell you what went wrong, and you will refill the object from the beginning. After you finish your input, it will print back a formatted chart of all the data you just entered for that person. Unformatted data currently stored inside the different class objects will be printed to the console for if the quick summary option is chosen.

**Description of Classes**

The person class is an abstract base class. It is inherited by both the Student class and the Teacher class. The student class is further inherited by the Athlete class. The Experience class is a composite class that exists in the Teacher class. All classes contain setters and getters for their private variables. All classes that inherit from other classes override any virtual functions. All classes have the << and >> operators overloaded to handle input and output of the class objects. The << operators display full formatted output and the >> operators will prompt the user for all data needed.

The Person class is designed to only be used for inheritance. It holds the values “firstName” and “lastName”, as strings, and “age”, as an integer. It also contains two pure virtual functions “classInfo()” and “quickPrint()”. “classInfo()” is used to print the characteristics of each child class. For example, a parent object (although not possible due to the class being abstract) would only be considered a parent. A student object would be considered a person and a student. “quickPrint()” is a function used to print compact, unformatted data when requesting a summary of what objects in the “SchoolPersonTester.cpp” file are filled with data.

The Student class inherits from the Person class as public. This class is meant to hold data for a student that is not an athlete. It adds four new variables of its own: “gradYear” and “currYear” as integers, “gpa” as a float, and “classes” as a string pointer that is used as a dynamic array. “gradYear” is used to hold the expected graduation date and “currYear” is used to hold the student’s grade level. “Gpa” holds the student’s gpa and the classes array stores each class the student is taking. The “gradYear” variable is initialized as -1 to handle quick printing with no data being entered into the object.

The Athlete class inherits from the Student class as public. This class is meant to hold data for a student that is also an athlete. It adds four new variables of its own: “jerseyNum” as an integer, “sport” and “position” as strings, and “scholarship” as a bool. The “sport”, “position”, and “jerseyNum” variables are self explanatory. The “scholarship” variable determines if the player is or is not playing on a scholarship. The “jerseyNum” variable is initialized as -1 to handle quick printing with no data being entered into the object.

The Experience class is a separate composite class that is used as an object in the Teacher class. It contains “degreeFocus” and “degreeLevel” as strings and “totalYears” and “currSchool” as integers. “degreeLevel” and “degreeFocus” are used to store the degree type (associate’s, bachelor’s, etc) and the area of study (computer science, creative writing, etc) respectively. The “currSchool” and “totalYears” variables hold the values for how long the teacher has taught at just their current school and how many years teaching total respectively.

The Teacher class inherits from the Person class as public. This class is the only template class.It is meant to hold data for a teacher. It adds four new variables of its own: “subject” as a string, “classNum” as an integer, “exp” as an object of type Experience, and “contact” that is a template type. “Subject” is the subject that the teacher teaches, “ClassNum” is the classroom number the teacher teaches at. “Exp” is the Experience object that contains the teacher’s education and prior experience. “Contact” contains the teachers contact information, either as an email, or a phone number. If the email contact is chosen, a Teacher<string> object is used. If a phone number is chosen, a Teacher<long long> is chosen. The “subject” variable is initialized as -1 to handle quick printing with no data being entered into the object.