Inheritance Programming Project

For this assignment, you are to create three classes. One is a parent class and the other two will be its children. The parent class is the "Person" class. The two children are the "Instructor" and "Student" classes.

Person Class

You will not actually instantiate objects of type person. This class is only used to define the common aspects of the "Instructor" and "Student" classes.

Class Variables

Variable Name	Variable Type
First Name	String
Last Name	String
Age	Integer
Lnumber	String

Class Methods

Default Constructor – create a person with Age = 0 and all other values being the empty string "";
Overloaded Constructor – take four arguments: First Name, Last Name, Age, LNumber
Setters – one for each variable.
Getters – one for each variable.

GetJob – returns "Undefined". Note, this does not access a variable, it simply returns a hard coded string.

Instructor Class

Inherits from Person.

Instructor has an additional class variable – Office Hours. This must be set in the constructors and you will need a get and set method for it.

GetJob – returns "Instructor". Again, this is not returning the value of a variable, it is simply returning a hard coded string.

Student Class

Inherits from Person

Student has an additional class variable – GPA – Float – display as X.XX. You will need methods to get and set GPA.

GetJob – returns "Student". As above, this is retuning a hard coded string.

Test Driver

There is a test driver in Moodle. It is set up so that you can separately test your Instructor and Student classes by commenting and uncommenting certain lines of code. It will create some instructors and students and make sure they behave as expected. It will also use polymorphism to test the values that are inherited from person

Review Criteria

The following are the criteria that you are to use for your peer review and that I will use for grading.

Criteria	Expected Results	Pass/No
Inheritance	Children do not duplicate parent variables or methods	
	Children properly utilize parent variables and methods	
	Proper use of Private/Protected/Public	
	Proper use of overwriting method	
	Proper adding of new variables and methods	
Design	Programs broken into three .h files and three .cpp files	
	Comments used to explain code	
Works	Program compiles and runs with test routine	
	Output matches that expected from test routine	