**Take-Home: Quiz 7 (15 pts) – Inheritance in C++**

Using Canvas <https://canvas.wsu.edu/>, please submit your solution to the correct quiz folder. Your solution should be a .pdf file with the name <your last name>\_quiz7.pdf and uploaded. To upload your solution, please navigate to your correct Canvas ***lab*** course space. Select the “Assignments” link in the main left menu bar. Navigate to the correct quiz submission folder. Click the “Start Assignment” button. Click the “Upload File” button. Choose the appropriate .pdf file with your solution. Finally, click the “Submit Assignment” button.

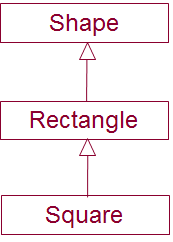
***Part I: Short Answer.***

**1. (5 pts)** What is inheritance? Explain.

Inheritance is one of the four pillars in OOP, which allows for creating a new class from an existing class. This derived class may acquire attributes and operations of the base class. This is really useful to implement many classes with some basic attributes in common. For example, we have a base class called Animal, then we can use inheritance to create classes like Dog and Cat. Dog and Cat classes have some basic attributes of Animal class; however, they may still have their own ones like Dog can swim and Cat eats fish.

***Part II: Fill-In-The-Blank.***

**2. (2 pts)** The following class diagram is an example of \_\_\_\_multilevel\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ inheritance.



**3. (2 pts)** Inheritance is representative of a(n) \_\_\_\_\_\_\_\_is-a\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ relationship. A Rectangle object demonstrates this relationship because it can be treated as a Shape object.

**4. (2 pts)** When an object of a derived class is instantiated, the base class’ \_\_\_\_\_\_\_constructor\_\_\_\_\_\_\_\_\_\_\_ is called implicitly or explicitly to initialize the data members of the base-class in the derived-class object.

**5. (2 pts)** \_\_\_\_\_Multiple\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ inheritance is demonstrated when a derived class inherits properties from 2 or more base classes.

**6. (2 pts)** Software \_\_\_\_\_\_\_reuse\_\_\_\_\_\_\_\_\_\_\_\_\_\_ provides higher levels of developer production through leveraging well tested, quality code.