CptS 122 – Data Structures
April 5, 2024

Your Name: _____ TA's Name: _____
ID#: _____ Section #: _____

SOLUTION: Take-Home: Quiz 8 (15 pts) – More OOP!

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>_quiz8.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.

1. **(8 pts – 2 pts/each)** What are the Four Pillars of Object-Oriented (OOP) ? Explain each using your own words.

1. **Abstraction:**

   Abstraction **is the concept that only essential information is presented to the outside word, while the background details are hidden. There are many levels of abstraction.**

   Procedural abstraction **refers to a sequence of steps/instructions that have a specific and limited function. The name of the procedure implies these functions, but suppresses the specific details. An example would be the word "open" for a door. (Pressman – Software Engineering)**

   Data abstraction **is a named collection of data that describes a data object. An example would be the data abstraction "door". The "door" would represent a set of attributes like door type, weight, etc. (Pressman – Software Engineering)**

2. **Encapsulation:**

   The grouping of data or attributes and the operations or functions that can be applied to the data or attributes. Encapsulation provides the ability to restrict access to the data and hide information.

3. **Inheritance:**

   The ability of a class (derived) to derive properties from a previously defined (base) class. The derived class customizes the properties of the base or acquired class. It's considered a form of software or code reuse.

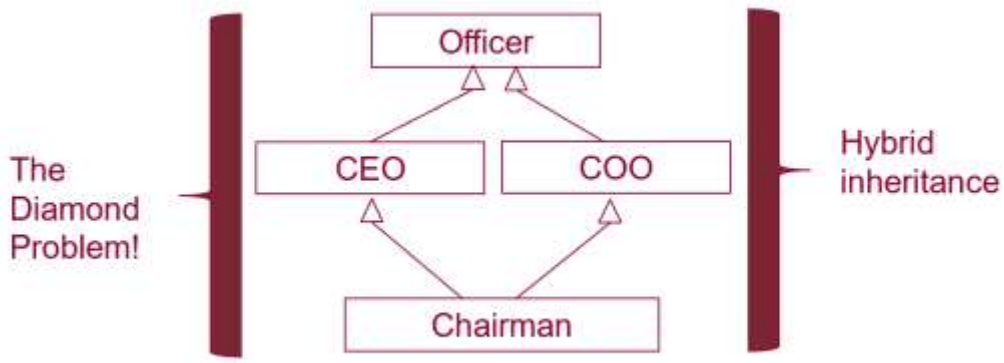
4. **Polymorphism:**

   Polymorphism **is the ability to use the same expression to denote different operations.** ***Runtime polymorphism*** **is the**

Your Name: _____ TA's Name: _____
ID#: _____ Section #: _____

ability to associate multiple meanings to a single function name though the use of late or dynamic binding. You can process objects of the same class hierarchy as if they are all objects of the hierarchy's *base* class. *Compile time polymorphism* is the type that is achieved through function overloading, operator overloading, and templates. Polymorphism enables you to "program in the general", instead of "program in the specific". Another form is *parametric* polymorphism the (data) type is left unspecified and later instantiated templates provide parametric polymorphism.

2. **(4 pts)** In regards to inheritance, what is the *diamond* problem? Explain.



The diagram above illustrates the diamond problem. A Chairman inherits from both CEO (Chief Executive Officer) and COO (Chief Operations Officer) classes, and CEO and COO inherit from the Officer class; it forms a diamond relationship. Here the "diamond" problem occurs because CEO and COO inherit from Officer, which have their own copies of the data members and methods acquired from the Office class. The Chairman class contains two subobjects - there is ambiguity in which members are accessed by Chairman.

3. **(3 pts)** What is a virtual function in C++? Explain.

A *virtual* function is a function whose behavior can be *overridden* or replaced. Function *overriding* is a feature that allows a derived class to provide a specific implementation for a function that is provided by a base class - this is NOT the same as function *overloading* - the return type, name, and parameters are the same in the base and derived classes