CS 162 Worksheet 9

Virtual vs. pure virtual vs. non-virtual functions

Recall:

Virtual

- Use virtual functions in a base class when you want to allow derived classes to override the implementation of the function if needed. The function in the base class <u>can have a default implementation.</u>

Pure virtual

- Use pure virtual functions in an abstract base class when you want to define an interface that must be implemented by derived classes. These functions <u>have no implementation in the base class and must be overridden by derived classes.</u>

Non-virtual

- Use non-virtual functions when you want to provide a specific implementation of a function in the base class that <u>should not be overridden</u> by derived classes.

1. Given the following classes and functionalities, list which functions should be virtual, pure virtual, or non-virtual:
   - Base class: Shape
   - Child classes: Triangle, Circle
   - Common variables: color, name
   - Unique variables to Triangle: base, height
   - Unique variable to Circle: radius
   - Functionalities:
     - Print the name
     - Print shape info (including name, color, and other variables)
     - Calculate area (each shape has a different way of calculating area)

Recall from last worksheet, Cardgame is the parent class, and Gofish inherited from it.

Now, assuming three other classes, Solitaire, Blackjack, and Rummy, are inherited from Cardgame.

1. How would you handle this so that all objects (Gofish, Solitaire, Blackjack, or Rummy) can be stored into a single array?

2. Which class can be abstract? What function(s) can be pure virtual, and why?

3. If any of the child classes contain dynamic memory that relies on destructor to free. What else needs to be done to avoid memory leaks?

STL:

1. What is the STL, and what are some things it provides to you?

2. What is the purpose of a map? When would it make sense to use one?

3. What are the differences between a vector and a set?

Open floor: Q&A for Assignment 4