Computer Science 2510 - Lab 5

Readings

- Class Notes
- Textbook: Chapter 14

Objectives

- To become familiar with inheritance and polymorphism.

Notes

- Most of the exercises in this lab were taken from the "Exercises" and "Drill" sections of Chapter 14 of the textbook (Bjarne Stroustrup, *Programming - Principles and Practice Using C++*, Second edition, Addison-Wesley, 2014, ISBN 978-0-321-99278-9.)

Lab Exercises

1. **Chapter 14 Drills**

   1.1. Define a class `B1` with a virtual function `vf()` and a non-virtual function `f()`. Define both of these functions within class `B1`. Implement each function to output its name (e.g., `B1::vf()`). Make the functions `public`. Make a `B1` object and call each function.

   1.2. Derive a class `D1` from `B1` and override `vf()`. Make a `D1` object and call `vf()` and `f()` for it.

   1.3. Define a reference to `B1` (a `B1&`) and initialize that to the `D1` object you just defined. Call `vf()` and `f()` for that reference.

   1.4. Now define a function called `f()` for `D1` and repeat 1-3. Explain the results.

   1.5. Add a pure virtual function called `pvf()` to `B1` and try to repeat 1-4. Explain the result.

   1.6. Define a class `D2` derived from `D1` and override `pvf()` in `D2`. Make an object of class `D2` and invoke `f()`, `vf()`, and `pvf()` for it.

   1.7. Define a class `B2` with a pure virtual function `pvf()`. Define a class `D21` with a `string` data member and a member function that overrides `pvf()`; `D21::pvf()` should output the value of the `string`. Define a class `D22` that is just like `D21` except that its data member is an `int`. Define a function `f()` that takes a `B2&` argument and calls `pvf()` for its argument. Call `f()` with a `D21` and a `D22`.

2. **Chapter 14 Review**

   2.1. What is a base class?

   2.2. What makes a class derived?

   2.3. What is a virtual function and how does it differ from a non-virtual function?

   2.4. What is the difference between a protected member and a private one?

   2.5. How does a pure virtual function differ from other virtual functions?

   2.6. Why would you make a member function virtual?

   2.7. Why would you make a virtual member function pure?

Author: Department of Computer Science, MUN (BE220531)