Lesson: Polymorphism in C++ version 2.0 Dr. Peter Liu

A. Overview

1. What is polymorphism?
   a) English: many forms
   b) analogy: a chameleon
   c) an hierarchy of classes: different types of objects (on the hierarchy) manifest different behaviors

```
             Employee
            ↑        ↑
        Manager    Staff
```

   - different behaviors: the pay calculation of a manager is different from that of a staff

   d) compile-time (static) data type of a variable/object vs
   run-time (dynamic) data type of an OBJECT

2. Why polymorphism?
   a) simplify coding
   - reusable and flexible coding

3. How to program polymorphism (inclusion polymorphism) in C++?
   a) use of a base-class pointer (or a base-class C++ reference parameter)
   b) virtual member functions

4. Key Concepts
   a) compile-time (static) function binding
   b) run-time (dynamic) function binding

B. Video Lessons

Part 1: Base-Class Pointers and Derived-Class Objects
https://ict.senecacollege.ca/~peter.liu/onlineS2020/Polymorphism/Polymorphism_Part1.mp4

Part 2: (Inclusion) Polymorphism and Virtual Functions
https://ict.senecacollege.ca/~peter.liu/onlineS2020/Polymorphism/Polymorphism_Part2.mp4

C. Code Examples

pay.cpp, payv.cpp

D. Course Notes

https://ict.senecacollege.ca/~btp200/pages/content/inclu.html
https://ict.senecacollege.ca/~oop244/pages/content/inclu.html