

SW코딩 역량을 위한 C++ Programming

- 2학년 1학기, 홍 봉희, 2017년
- 부산대학교 컴퓨터공학과

Contents

1. C++ 공부는
2. C++ 공부에 필요한 C 언어 중요 요소
3. 객체 지향 프로그래밍과 C++ 언어
4. C++ 언어의 요소들
5. Abstract Data Types, Classes and Objects
6. Functions of Class
7. Inheritance
8. Polymorphism and Dynamic Binding
9. Class Hierarchies and Abstract Classes
10. Genericity and Templates
11. Handling Exceptions and I/O Operations

1. C++ 공부는?

1.1 Who has never played with legos as a child

- by combining legos, you could easily create models of real objects
 - assemble components
 - build by using constructs
- object-oriented language
 - objects: like a box of legos
 - object-oriented programming: 객체를 결합하여(조립하여) 더 복잡한 문제를 해결하기 위한 프로그램을 개발

√

1.2 C++를 공부하기 위해서는?

- to learn C++, you need a basic knowledge of C language and some basic practice in programming
- The objectives of this course: to learn the skill of C++ programming for data structures and file structures
- three fundamental ways to use C++, according to Bjarne Stroustrup
 - as a better C: const, overloading, inline, reference
 - as a language supporting data abstraction
 - as a language supporting object-oriented programming

○ data abstraction

- refer to the ability to add (to a programming language) user-defined data types that describe abstract entities
- 데이터 추상화: 추상화는 프로세스 추상화와 데이터 추상화가 있다. 1)프로세스 추상화는 함수처럼 함수 body는 몰라도 함수 호출만으로 사용하는 것. 2)데이터 추상화는 데이터 구성에 대한 세세한 부분은 숨기하고 접근하지 못하게 하는 것.
- 추상 데이터 타입(ADT: Abstract Data Type): 데이터 타입은 구조, 연산, 제약조건을 모두 기술한 것이고 추상 데이터 타입은 타입 이름과 외부에서 사용될 공개 인터페이스로서 연산만 기술한 것. 보통 데이터 타입의 구현 부분이 빠진 것.

○ 객체 지향(Object-oriented)

독립적인 객체로 프로그램이나 시스템을 구성하는 것, 소프트웨어를 데이터 구조와 behavior가 결합된 객체들의 집합으로 보는 것

○ 객체(Object)

객체란 인간이 다른 것과 구분해서 인식할 수 있는 '어떤 대상'을 의미. 소프트웨어에서 객체는 그 내부에 속성(attribute)을 갖고 외부로부터 메시지(message)를 받아 처리

○ 엔티티(Entity)

- 인간이 생각하는 개념 또는 사물 또는 정보의

세계에서 의미 있는 단위

- 사전적 의미: 실체, 존재, 자주적인 것, 본질
- 객체 지향 프로그래밍(Object-Oriented Programming)
 - 객체라는 작은 단위로서 모든 처리를 표현. 프로그램을 객체들의 모임과 이들 간의 상호작용으로 보고 프로그래밍하는 것

1.3 The construction process: object-oriented programming

- use C++ in an object-oriented manner
 - need to use object-oriented programming
 - need to do object-oriented design
- simpler to make the transition from conventional programming to object-oriented programming
 - need to learn to distinguish those elements of C++ that make it a better C and a C with data abstraction
 - ✓ must be clearly related to the principles of object-oriented programming
- constructs
 - programming에 사용되는 문장, 명령어, 키워드

- 프로그램의 제어 구조인 sequence, selection, iteration 등을 construct
- 프로그램 구성 요소

1.4 O-O Programming language and program design method

- 프로그램 설계 기술이 중요
 - 프로그래밍 언어가 프로그래머의 생각을 지배
- 객체 지향 프로그래밍과 객체 지향 설계
 - 정보 숨김: 프로그램 설계 방법론의 기본 원칙
 - √ Encapsulation: 어떤 하나를 다른 것에 포함시킬 때 포함된 것이 외부에서 보이지 않게 하는 것
 - √ public interface
 - √
 - 추상화: 프로그래밍 언어의 데이터 타입을 정의하는 것과 관련됨
- reuse
 - C 언어의 library 사용과 C++ 언어의 class library 사용 중에서 어느 것이 더 좋은가?
 - √ C programming: top-down functional decomposition
 - √ C++ class library
 - >>소스 코드를 수정하지 않고도 추상화된 데이터 타입에 새로운 함수의 추가 또는 재정의 가능
 - >>다른 프로그래머에 의해 개발된 코드의 재사용 가능