



1장 C++ 기초

최 광 훈

전남대학교 전자컴퓨터공학부
소프트웨어공학전공



C++개요

□ 객체지향 프로그래밍 방법

- 데스크탑에서 모바일까지 다양한 응용 사례
 - Windows MFC (C++), Linux Qt (C++)
 - iPhone OS (Objective-C), Android (Java), Windows Phone (C#), Tizen(C++)
- 소프트웨어 엔지니어 job의 주요 자격 요건
 - E.g., 구글: “Strong object-oriented skills”, “Extensive programming experience in C++/Java”
 - E.g., 넥슨: [공통요건] C/C++ 프로그래밍 언어를 능숙하게 사용할 수 있는 분. Visual Studio 개발 환경을 능숙하게 사용할 수 있는 분. 자료구조, 알고리즘에 대해 이해하고 있는 논리적인 사고 능력이 있는 분. ...

□ 객체지향 프로그래밍 방법이 성공한 이유?

- (보조 강의 자료 참고)

C++ 개요

□ C++언어의 기원

- 1960년대 LISP언어에서 클래스나 상속과 같은 객체지향 개념을 인공지능 분야에 응용
- 프로그래밍 언어에서 객체지향 개념을 도입함
 - Simula (Ole-Johan Dahl & Kristen Nygaard, 1962년)
 - Smalltalk (Alan Kay, 1980년)
- 1983년 Bjarne Stroustrup (AT&T)가 C언어에 객체지향 개념을 추가하고, 타입 검사를 비롯하여 여러 기능을 개선하여 C++를 설계
 - The C++ Programming Language 책의 1st ed. (1985년).
- ANSI/ISO C++ 표준
 - Special Edition (또는 3rd ed.) (2000년)





C++ 개요

□ C++언어에서 지원하는 프로그램 설계 원리

- 모듈화 => 추상 데이터 타입(Abstract Data Type)
 - 클래스와 접근 권한(private, public, protected)
 - 주어진 연산들만을 사용해서 데이터를 사용할 수 있고, 그 외의 방법으로 접근하는 코드를 쓰면 컴파일 에러
- (객체 지향적) 재사용 => 클래스 상속
 - 클래스 상속
 - 새로 만들려는 것과 유사한 기존의 클래스를 찾아 일부만 수정/추가하여 새로운 클래스를 만드는 방법
- (타입 기반, 비객체 지향적) 재사용 => 템플릿
 - 템플릿
 - 동일한 코드를 여러 데이터 타입에 대해 사용하는 방법