

C++ Programming and Practice

2024년 Spring

교수소개

- 김태연
 - 부산대학교 소프트웨어교육센터
 - 소프트웨어 품질 연구실
 - 자연대연구실험동 305호, ☎3940, tykim@pusan.ac.kr
 - 문의: 메일, PLATO 쪽지, Q&A 게시판 등
- 주요 연구분야
 - 소프트웨어 공학
 - 소프트웨어 테스트
- 약력
 - 부산대학교 소프트웨어 품질 연구실 박사수료
 - 부산대학교 정보 컴퓨터공학과 98학번
 - LG전자 모바일 사업부 - 안드로이드 휴대폰 개발
 - 인사이드 안드로이드, 안드로이드 미디어 프레임워크 저술
 - 스타트업 창업 및 대표이사 역임, 투자유치, 50억 매출 달성

학습목표

- 객체지향 개념에 바탕을 둔 Modern C++ 프로그래밍
 - S/W 개발을 전문 직업으로 선택한 학생들의 경력 토대를 마련
 - C++ 프로그래밍의 기본 개념을 학습
 - C++ 언어를 통해 객체지향 개념을 학습
 - C++ 프로그램의 설계 능력과 구현 능력을 기름
 - C++ 도구를 사용한 디버깅 및 개발 능력을 체득함
 - 코드를 기반으로 동료들과 커뮤니케이션 및 협력하는 법 훈련
 - C++11, C++14, C++17 등의 표준을 따르는 Modern C++ 학습

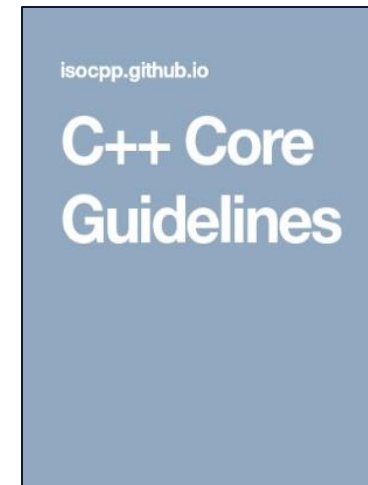
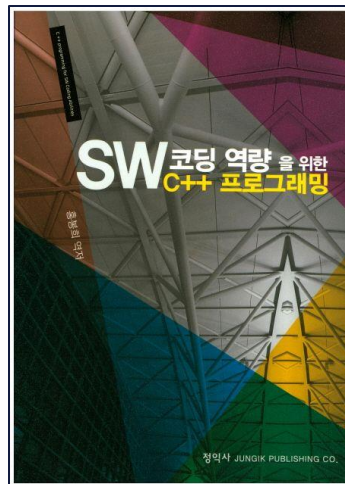
Modern C++ Programming*

- 현재 ISO C++ 표준을 사용한 모범 사례(best practices)
- 완전한 형식 안전(type-safe) 및 리소스 안전(resource-safe) 코드를 목표로 함
 - static type safety, well-specified interfaces
 - resource safety, constructors/destructors, RAII
 - abstraction, often zero overhead
 - encapsulation, invariants, classes
 - generic programming, templates
 - simplicity for most developers, complexity hidden in libraries
- C++ Core Guidelines

교재

- 강의자료 (시험 시 출력하여 지참 가능)
- 교재 (시험 시 지참 가능)
 - 비타민 C++, 채흥석 교수님 (절판)
 - SW 코딩 역량을 위한 C++ 프로그래밍, 홍봉희 교수님
- 참고자료
 - C++ Core Guidelines

§ <http://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines.html>



수업 일정

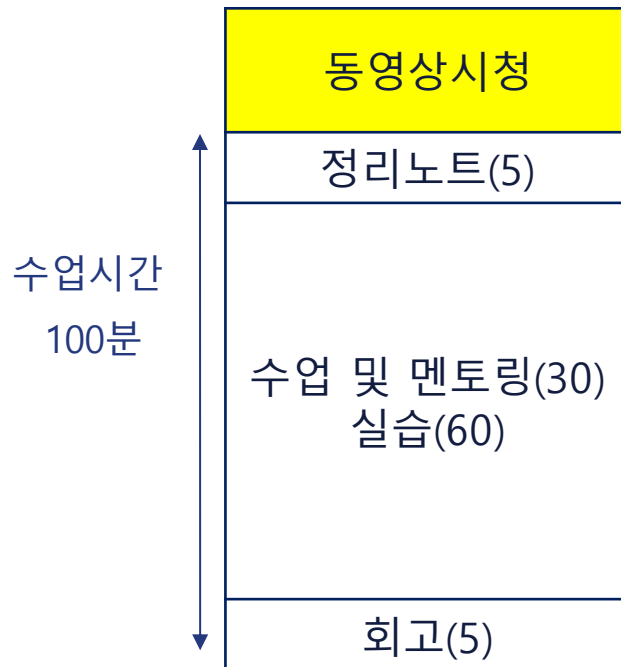
주차	범주	내용	Lab
1	Better than C	타입 - 변수, smart pointer, auto	
2		stl – array, vector, string, pair, map	
3		control structure	v
4		function - <algorithm>, std::function, lambda	
5	Classes	클래스	
6		default operators - constructor, copy, destruct or	
7		객체 지향적 사고 1	
8		중간고사 (Lab 대체)	v
9	Classes: Inheritance & Polymorphism	연산자 오버로딩	
10		상속	
11		다형성	
12		객체 지향적 사고 2	v
13	Large SW Development	템플릿	
14		예외 처리	
15		보충주간	
16		기말고사 (Lab 대체)	v

LAB 일정

	월	화	수	목		월	화	수	목
3/4		0 소개		1.2 포인터	4/29		7 00 1		7 00 2
3/11		2 STL 1		2 STL 2	5/6	휴무	9 상속 1		9 상속 2
3/18		3 제어문		4 함수 1	5/13		10 다형성1	휴무	10 다형성2
3/25		4 함수 2		LAB 1	5/20		OOT 2		LAB 3
4/1		5 클래스1		5 클래스2	5/27		11 템플릿1		11 예외1
4/8		6 DO 1	휴무	6 Do 2	6/3		11 예외2		휴무
4/15		7 OOT 1		7 OOT 1	6/10		보강		보강
4/22		중간고사		-	6/17		기말고사		-

수업진행 방식

- 수업일자
 - 화, 목: 이론 및 실기
 - 강의 동영상 제공 - <https://www.edwith.org/pnu-modernC>
- 과목 게시판
 - <http://plato.pusan.ac.kr>



- 영상 내용 3줄 요약
- 권장 - 수업마다 하나씩
- 필수 - 매주 하나씩 (체크함)
- 실습 문제 중 각자 레벨에 맞춰 훈련
- 짝지와 페어 혹은 멘토링 등 실시
- 좋은 점, 아쉬운 점, 아하! 한 줄 씩
- 권장 - 수업마다 하나씩
- 필수 - 매주 하나씩 (체크함)

학습영상 시청 및 핵심 내용 정리

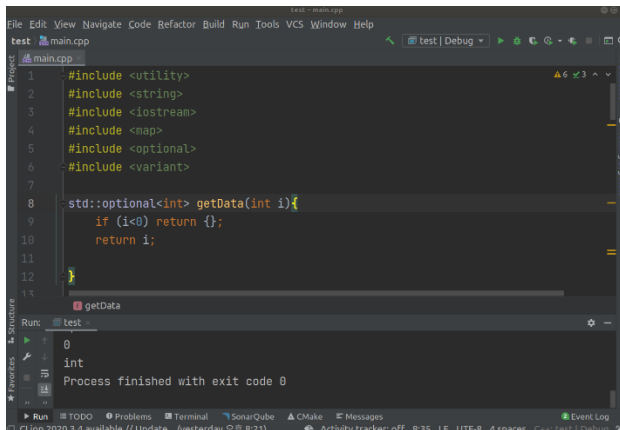
- 학습 영상은 스스로 시청하며 공부함
- 학습 영상은 수업 전 시청을 목표로 함
- **학습 영상 시청 후 핵심 내용을 정리하여 정리노트 게시판에 제출**
 - 학습 영상을 시청하면서 중요하거나 흥미롭다고 생각하는 요점을 정리함

요점 정리 예

1. 람다식을 사용하여 간단한 함수를 구현할 수 있다.
2. extern을 사용하여 다른 코드의 변수를 사용할 수 있다.
3. std::accumulate 함수를 사용하면 배열에서 계산을 쉽게 할 수 있다.

실습 및 과제

- 실습/과제 결과 제출은 대부분 “*PLATO 코딩과제*”로 제출 가능함
- 실습 문제는 자신이 할 수 있는 문제 위주로 도전
- 과제는 Lab 을 진행하기 위한 필수 요소
- **부정적인 방법 (예, 출력 값만 일치, 표절, 로그 조작 등) 은 마이너스 점수**
 - 우리 학과 방침: ChatGPT를 학습에 활용 가능하나 **표절에 주의***



```
#include <utility>
#include <string>
#include <iostream>
#include <map>
#include <optional>
#include <variant>

std::optional<int> getData(int i){
    if (i<0) return {};
    return i;
}
```

Run: test

0
int
Process finished with exit code 0



```
2021-06-19T18:21:49.542+09:00,edward,KeyEvent,61:
2021-06-19T18:21:49.631+09:00,edward,KeyEvent,32:
2021-06-19T18:21:53.060+09:00,edward,KeyEvent,97:
2021-06-19T18:21:53.172+09:00,edward,KeyEvent,117
2021-06-19T18:21:53.311+09:00,edward,KeyEvent,116
2021-06-19T18:21:53.390+09:00,edward,KeyEvent,111
2021-06-19T18:21:53.478+09:00,edward,KeyEvent,32:
2021-06-19T18:21:55.975+09:00,edward>Action,Edito
2021-06-19T18:21:55.977+09:00,edward,KeyEvent,10:
2021-06-19T18:21:57.563+09:00,edward,KeyEvent,114
2021-06-19T18:21:57.662+09:00,edward,KeyEvent,101
```



CLion 및 ActivityTracker 플러그인

온라인 로그 분석 및
표절 검사 (JPlag)

수업현실

- 이전 수업들 (강의 경험)
 - 태권도 흰띠와 검은띠가 함께 수업을 진행함
 - 백준 기준으로 브론즈5와 플래티넘4가 함께 실습함
 - 태권도 1단, 2단, 3단, 4단 을 한학기에 마스터 하려함
 - 태권도, 수영 등은 한 학기 수업으로 마스터 할 수 있는가?
 - 수영장에 들어가지 않고 수영을 동영상과 책으로 공부하고 시험 출전
 - 코딩과 컴파일을 해보지 않고 Lab/시험 보러 옴
- 학생들*
 - 많은 학생들이 예상하지 못한 것을 거부합니다.
 - 많은 학생들이 시험을 위해 공부하려고 합니다.
 - 많은 학생들이 명시적으로 언급된 전제 조건조차 무시합니다.
 - 많은 학생들이 교수보다 "웹"을 더 신뢰합니다.

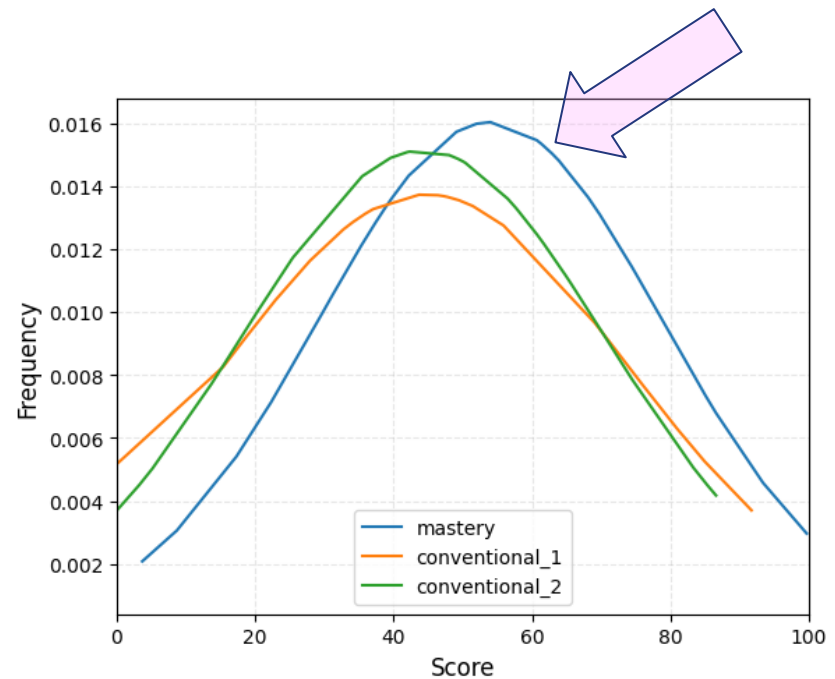
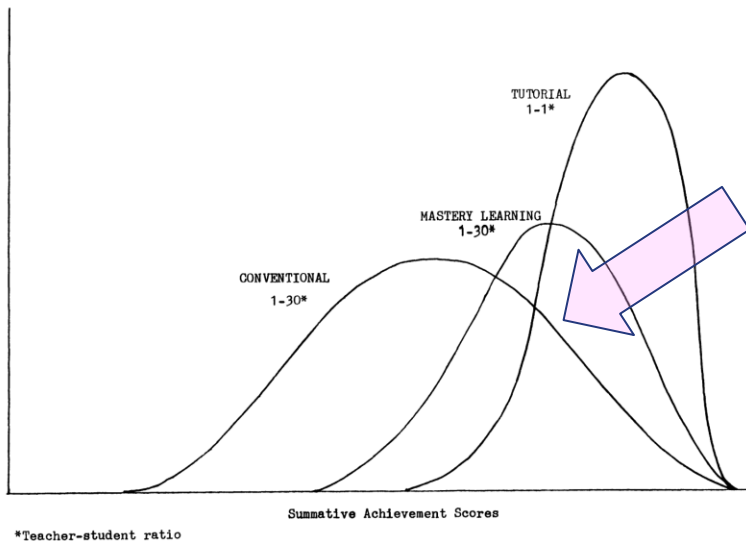
학습 방법들

- Mastery Learning (숙달 학습)*
 - Students **must achieve a level of mastery** in prerequisite knowledge before moving forward to learn subsequent information.
 - The focus of instruction **should be the time required** for different students to learn the same material and achieve the same level of mastery
- Paired Learning (짝지와 함께 학습하기)
 - Lab1 / Lab4 이후 짝을 정함
 - 짝은 서로 함께 도와 가며 수업 시간에 실습/과제 문제를 해결함
 - 서로 끝까지 수업을 완주할 수 있도록 코칭, 동기부여 및 격려와 응원하기
 - 짝이 80% 이상 출석 시 보너스 2점 부여함 (중간/기말 각각 평가)
- 튜터링
 - 서로에 대한 존중과 배려의 의미로 호칭은 서로 OO님으로 하고 서로 존대함
 - 시작 후 처음에는 해당 문제를 읽고, 서로 의견을 나눠 봄
 - 해결해야 할 일들을 대략 To Do List에 적어 봄
 - 수강생은 튜터를 신뢰하며 대화하고 질문하면서 문제를 어떻게 풀어야 하는지를 파악
 - 튜터는 수강생을 존중하며, 상대에게 적절한 질문도 하고, CLion 사용법도 알려 주면서, 수강생이 자신의 해결책을 도출하고 이를 코딩할 수 있도록 도와 줌

그동안의 성과

- 각자의 레벨에서 의미 있는 수업 진행 방식을 고민함
 - 2021년 부터 9번의 프로그래밍 수업으로부터 피드백을 반영하여 v1.0
 - v0.3 (2021년 2학기) 부터 학생들로부터 의미 있는 피드백이 전달 되었음
 - v0.7 (2022년 2학기) 수업평가 개인 최고점수 갱신 (공대 평균에 근접)
 - v0.9 (2023년 1학기) 학습성과 평균 50점 돌파 (100점 만점)

FIGURE 1. Achievement distribution for students under conventional, mastery learning, and tutorial instruction.



숙달학습 피드백

- 수준별 진도
 - 학생 수준에 맞춰서 진행해 주셔서 감사합니다.
 - 학생 개인별 진도 지도에 있어서 흥미를 잃지 않게 지속한다는 점의 수업방식에 훌륭하다고 느꼈습니다.
 - 중간고사 성적을 바탕으로 나누어서 각자 진도에 맞게 수업하는 것이 좋았습니다. 저 같은 경우에는, 클래스를 다루기 시작하면서, 점점 따라가기 벅차다고 느꼈습니다. 그때부터, 자신의 실력에 맞추어서 수업을 진행하니 마음을 조금하게 먹지 않아도 되어서 부담감이 줄었습니다.
 - 중간고사 통과를 못한 소위 뒤처지는 학생들을 끝까지 챙겨주는, 가령 중간 이후 학기초 주차로 돌아가서 다시 실습을 풀 수 있게 하는 것이 나로 하여금 계속 따라갈 수 있게 해 주었다.

튜터링 피드백

- 1:1 과외다!
 - 다른 수업을 들으면 모르는 것이 생겼을 때 주변 친구들이나 교수님에게 물어보는 것도 한계가 있어 자주 포기하곤 했는데, 이 수업의 튜터링은 거의 1:1로 나를 전담을 해서 내가 헤맸던 부분이나 추가로 궁금했던 부분을 유연하게 알려주고 배울 수 있는 시간이라 정말로 유익했음
 - 튜터링을 통해 이미 실습, 과제를 해결한 튜터분들께 1:1로 질문할 수 있어서 좋았고 내 코드를 꼼꼼히 짚으며 문제를 피드백하는 경험은 특별했으며 덕분에 강의자료의 예시와 다른 경우에 활용하는 방법을 배울 수 있었고 이게 이해하는 데에 큰 도움이 되었음
- 가르치는 것 뿐만 아니라 배우는 것도 있다!
 - 튜터링 짝을 맺어줄 때 꼭 배우는 입장에서만 할 필요는 없다고 말씀해 주셔서 용기를 내어 3주차부터 튜터로 과제를 끝내지 못한 분을 가르쳐 봤는데, 주차가 점점 지나갈수록 튜터링을 많이 참여하게 되고 C++에 더 자신감이 붙어서 이제는 프로그래밍이 더 재미있어졌음
 - 튜터링 수업을 통해 튜터로써는 제가 맡은 수강생들이 과제의 의도를 정확히 이해하고 제출할 수 있도록 하기 위해 튜터링 이전 30분동안 수업 준비를 하고, 수강생으로서는 정말 제가 이해가 안 되는 부분이나 효율적인 프로그래밍을 배우기 위해서 노력하면서 더욱 이해도가 높아졌음
 - 타과생으로서 늘 혼자 코딩하다가 막히면 구글에 검색하기만 했는데 튜터링을 하고, 코드를 서로 봐 주고 조언하는 과정에서 가르치는 것도, 배우는 것도 많은 공부가 된다는 것을 느꼈음
- 튜터링 참여의 목적은 다양하다!
 - 튜터링 중 상대방과 나 사이에 어떤 생각의 차이가 있는지, 그리고 그걸 좁히는 과정을 경험했고, 앞으로도 프로그래밍에 대한 소통을 하게 될 텐데, 내 코드를 설명하는 방법 등을 알아감
 - 전공생 분들에게 많이 배워보자는 생각을 가지고 튜터링 시스템을 이용하겠다고 생각함
 - 코딩 실습과 튜터링은 입사 코딩 테스트와 면접 같은 관계에 있다고 생각함

수업을 통해 얻을 수 있는 것

- 코드 디자인 및 읽는 능력의 향상
 - 지금까지는 하나의 파일에 작성된 코드들이 많았고, 팀 역시 팀원들의 파라미터 정도만 신경 썼음
 - 실습/과제 코드를 받으면 이 코드들이 어떻게 동작하는지 분석함
 - main()에서 부터 한 줄씩 코드를 읽는 습관을 가지고자 노력함
 - 강의 영상은 코드를 리뷰한다고 생각하고 시청함
 - 전체적인 그림을 그려보고 코드를 읽는 방식이 매우 효과적이었음
 - 상속, public, private, 객체 지향 코드 등이 이해가 잘 되며 코드 작성 및 읽기 능력이 향상됨
- 다양한 실습/과제 경험
 - PLATO 의 코딩 과제를 통해 실습/과제를 진행하고 실시간으로 결과를 확인할 수 있었음
 - 매주 배운 내용을 쉽게 따라해 볼 수 있는 소과제인 실습의 존재가 좋았음
 - 실습을 직접 해 봄으로써 다시 영상을 시청할 때 이해가 더 잘됨
 - 모르는 부분은 영상을 돌려보며 찾아서 시청함
 - 실습/과제를 해결한 경험이 쌓일수록 다른 과목의 과제 난이도가 두렵지 않음
 - 방학을 이용해서 수업시간에 배운 내용으로 개인 프로젝트를 시작
- 수업 활동을 통한 성장
 - 어떤 것에 도전하고 싶다면 망설이는 것보다 시도해 보는 것도 나쁘지 않은 것 같음
 - 다시 영상을 시청할 때 내가 쓴 회고를 읽고 좋았거나 깨달음이 있는 부분을 집중적으로 시청함
 - 수업을 듣는 다른 사람에게 튜터링을 해 준 경험은 생각보다 괜찮았음
 - 한 학기 동안 많은 시간을 투자하였고 그만큼 얻어가는 것도 많은 수업이었음

평가방법

- 평가방식
 - 중간고사, 기말고사, Lab 60% (Lab 4번)
 - 출석: -2점, PLATO 스마트 출석부 (2번) + 영상/정리노트/회고
 - 실습: 15%
 - 과제: 15%
 - 동료지도: 10%
 - 보너스 점수 - 짝 출석
- 기타
 - 출석: 결석 2점 감점
 - 1/3 이상 결석 시 - F
 - 정당한 사유 없이 Lab/시험 불참 시 - F
 - 주의: 선수과목 C 언어 (1학년 2학기 과목) 경험 필요

Q&A
