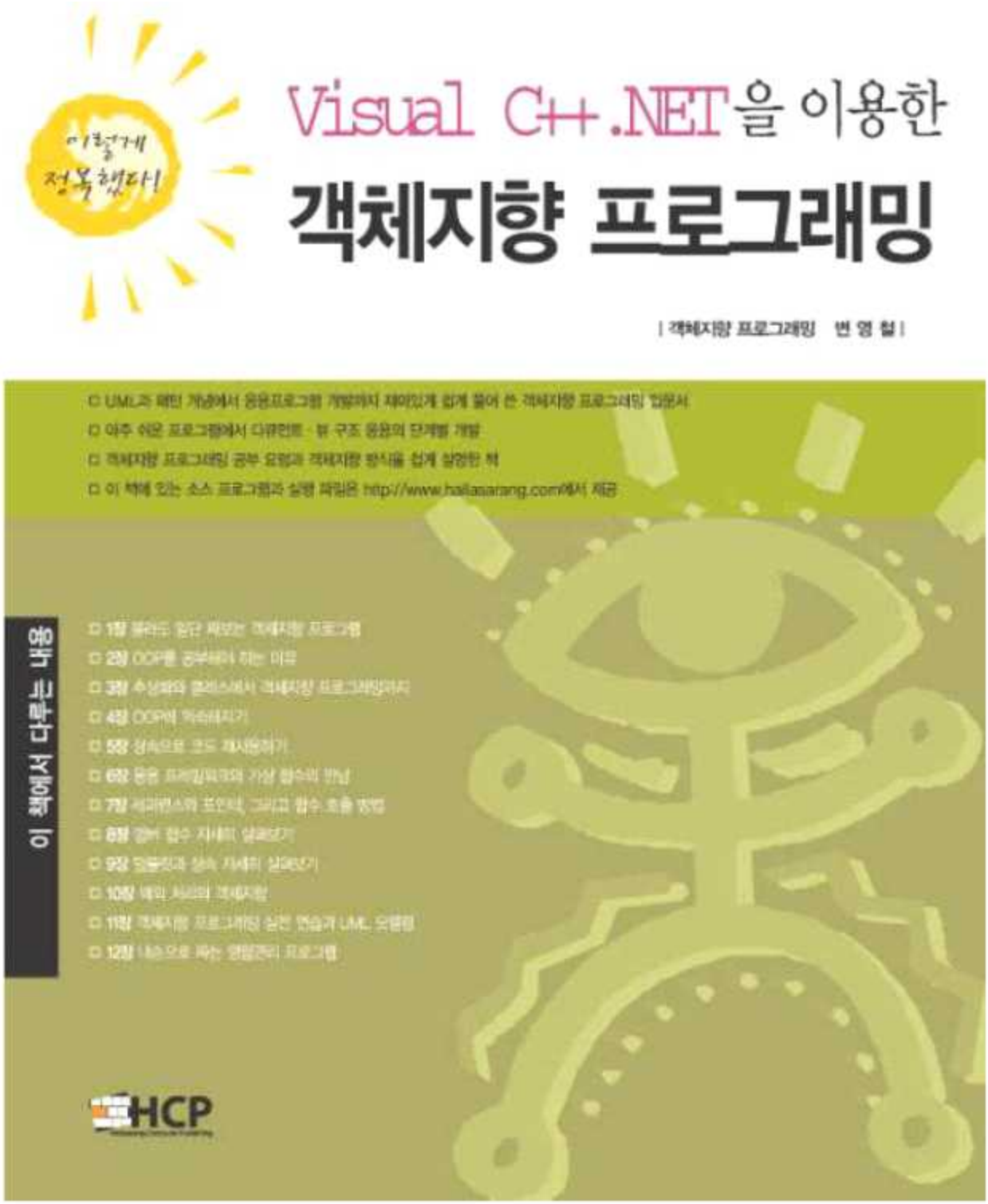
객체지향 프로그래밍(C++) 교재 (구내서점)



1. 본 강의에 대한 간단한 소개

여러분, 어제 무엇을 하셨나요?

제가 대답을 해 보겠습니다. "저는 어제 집에서 쉬었습니다." 만일, 이 대답이 이상하지 않다고 생각하면 여러분은 이미 객체지향 개념의 절반을 알고 있는 것과 같습니다.

집에서 쉬었다고 했지만, 사실은 쉬기만 했을까요? TV도 보고, 전화도 하고, 식사도 했을 것입니다. 여러 일을 했지만 간단하게 한 줄로 표현하였습니다.

복잡한 것을 묶어서 간단하게 표현하는 것, 이런 것을 추상화(abstraction)라고 합니다. 코드를 간단하게 표현하면 함수가 되고, 데이터와 코드를 같이 묶어 간단하게 표현하면 클래스가 됩니다. 추상화를 알면 객체지향 프로그래밍(OOP)은 다 알고 있는 것이나 다름없습니다.

여러분은 이미 객체지향 개념을 이해하고 있고, 객체지향 프로그래밍을 쉽게 정복할 수 있습니다. 하지만 그런데도 OOP를 어렵다고 생각하는 학생들이 있습니다. 다 알고 있으면서 어렵게 생각한다? 무엇이 문제일까요?

바로 습관화가 답입니다. 객체지향 방식의 생활에는 너무나도 익숙해 있으면서 객체지향 방식의 프로그래밍에는 전혀 익숙해 있지 않기 때문에 어렵다고 느끼는 것입니다. 객체지향 방식으로 생각하고 객체지향 방식으로 프로그래밍하기! 이 강의를 통해 여러분이 객체지향 방식의 프로그래밍에 쉽게 익숙해질 수 있도록 설명할 것입니다.

또한, 실습실 수업과 별도로 온라인 동영상 강의 자료도 있으니 언제라도 예습과 복습을 통하여 OOP에 관해 공부하시기 바랍니다.

2. 강의 교재 및 강의 내용

강의 교재는 학생회관에 있는 구내서점에서 구매할 수 있습니다.

교재명 : Visual C++.NET을 이용한 객체지향 프로그래밍

지은이 : 변영철, 출판사 : HCP

장별 강의 내용은 다음과 같습니다.

제1장 • 몰라도 일단 짜보는 객체지향 프로그램 (동영상 강의 참고)

제2장 • OOP를 공부해야 하는 이유

제3장 • 추상화에서 객체지향 프로그래밍까지1

제3장 • 추상화에서 객체지향 프로그래밍까지2

제4장 • OOP에 익숙해지기

제5장 • 상속으로 코드 재사용하기

제6장 • 응용 프레임워크와 가상 함수의 만남1

제6장 • 응용 프레임워크와 가상 함수의 만남2

제6장 • 응용 프레임워크와 가상 함수의 만남3

제7장 • 레퍼런스와 포인터, 그리고 함수 호출 방법1

제7장 • 레퍼런스와 포인터, 그리고 함수 호출 방법2

제8장 • 멤버 함수 자세히 살펴보기

제9장 • 템플릿과 가상 함수 자세히 살펴보기

제10장 • 예외 처리

제11장 • 객체지향 프로그래밍 실전 연습과 UML 모델링 (동영상 강의 참고)

3. 대상 학생

이 강의는 객체지향 프로그램이 어렵다고 생각하는 사람, 한 번 정도는 포기하고 싶었던 생각을 가졌던 사람, 혹은 이제 객체지향 프로그래밍을 이제 막 시작하려는 사람을 위한 강의입니다. 따라서 이 강의는 객체지향 프로그래밍 초급자를 위한 것이기도 하고, 전체 내용 면에서는 초/중급자를 위한 내용도 있습니다. 이 강의는 C 언어에 대해 잘 모르더라도 들을 수 있습니다. C 프로그램을 작성해본 경험이 없더라도 이해하는 데 문제가 없습니다만 그러한 경우에는 C 언어 책을 옆에 두고 강의를 들으시기를 권합니다.

4. 장별 강의 내용

제1장 강의는 동영상 강의로 따라 하기 바랍니다. 따라서 본 강의는 2장부터 시작하겠습니다.

제2장 • OOP를 공부해야 하는 이유

객체지향 프로그래밍이 정말로 중요하고 꼭 필요하다는 걸 진심으로 느끼고 공부하는 사람이 몇이나 있을까요? 객체지향이 아닌 방식으로 프로그램을 많이 작성해 본 사람은 어느 정도 느끼겠지만요. 이 장에서는 왜 객체지향 프로그래밍을 배워야 하는지를 쉽고 간단하게 설명함으로써 객체지향 프로그래밍을 왜 공부해야 하는지 그 필요성에 대해 살펴보도록 하겠습니다.

제3장 • 추상화에서 객체지향 프로그래밍까지

추상화를 이해하면 클래스를 이해할 수 있고, 클래스를 이해하면 객체를 이해할 수 있습니다. 따라서 추상화를 이해하면 객체지향의 절반을 이해하는 것입니다. 사실 여러분은 매일 추상화를 하고 있을 정도로 추상화 개념은 쉽게 접할 수 있고 쉽게 이해할 수 있는 내용입니다. 이 장에서는 추상화에서부터 클래스까지 내용을 쉽게 익힙니다.

제4장 • OOP에 익숙해지기

객체지향 프로그래밍은 너무나도 자연스러워서 누구나 쉽게 이해할 수 있습니다. 하지만 실제로는 그렇지 못합니다. 왜일까요? 실제 생활은 객체지향 방식으로 하면서 프로그래밍을 할 때에는 여간하여서는 객체지향 방식으로 생각하지 못하기 때문이지요. 이 장에서는 객체지향 프로그래밍을 쉽게 정복하는데 도움이 되는 몇 가지 습관 혹은 철칙에 대해 공부합니다.

제5장 • 상속으로 코드 재사용하기

저는 상속이라는 말을 싫어합니다. 왜냐하면, 상속이라고 하면 왠지 자신도 모르게 부모와 자식 관계를 연상하게 되기 때문이지요. 객체지향 프로그래밍을 공부하면서 이런 생각은 독과 같습니다. 객체지향 프로그래밍에서의 상속은 부모와 자식 관계를 설정하기 위한 것이 아닙니다. 단순히 코드를 재사용하기 위한 것입니다. 이 장에서는 그러한 내용에 대해 살펴봅니다.

제6장 • 응용 프레임워크와 가상 함수의 만남

가상 함수는 객체지향 프로그램의 위력을 느낄 수 있도록 해주는 대단한 도구입니다. 막연히 가상 함수가 무엇인지를 아는 것은 중요하지 않습니다. 마음속 깊은 곳에서 아하 가상 함수가 이런 것이구나 하고 느낄 수 있어야 합니다. 이 장에서는 이제까지 공부한 내용을 토대로 무엇인가 다른 것을 느낄 수 있도록 할 것입니다.

제7장 • 레퍼런스와 포인터, 그리고 함수 호출 방법

7장에서 10장까지는 객체지향 프로그래밍을 잘하는데 도움이 되는 문법들을 설명하였습니다. 가급적 지루하지 않도록 질문을 하고 답을 하는 형식으로 정리하였습니다. 이 장에서는 레퍼런스와 포인터 위주로 왜 필요하고 언제 활용할 수 있는지를 중심으로 설명하겠습니다.

제8장 • 멤버 함수 자세히 살펴보기

클래스 및 멤버 함수와 관련된 내용 중에서 몇 가지 여러분이 알고 있어야 할 내용이 있습니다. 콜론 초기화, 친구 함수 및 클래스, 연산자 중복 정의, 디폴트 멤버 함수, 정적 멤버 함수 등이 그것이지요. 이 장도 문법적인 개념이 주를 이루기 때문에 될 수 있으면 지루하지 않도록 질문에 답을 하는 형식으로 구성하였습니다. 어떤 프로그래밍 언어를 공부하더라도 반드시 알고 있어야 할 내용을 다루므로 꼭 읽어봐야 합니다.

제9장 • 템플릿과 가상 함수 자세히 살펴보기

MFC 혹은 자바를 이용하여 프로그램을 작성하다 보면 컬렉션 클래스 템플릿이 볼 수 있는데, 이를 이용하면 데이터를 유지 관리하는 프로그램을 아주 쉽게 작성할 수 있습니다. 이 장에서는 템플릿을 직접 작성해 봄으로써 라이브러리로 제공되는 템플릿을 쉽게 사용할 수 있도록 도움을 줍니다. 또한, 앞 장에서 다루지 못한 가상 소멸자와 순수 가상 함수를 공부하고 왜 필요한지를 공부합니다.

제10장 • 예외 처리

예외 처리를 처리하는 것 그 자체보다도 예외와 관련된 코드를 데이터 추상화하여 클래스로 작성하고, 추가로 정보를 예외 처리 시 이용하는 방법에 대해 이해함으로써 데이터 추상화가 예외 처리에 어떻게 응용되는지 알기 쉽게 설명합니다. 이 장을 공부함으로써 이제 본격적으로 객체지향 프로그램을 개발할 준비가 됩니다.

2장에서 10장까지 강의가 끝난 후 이제 여러분은 제1장과 제11장을 공부할 수 있습니다. 강의시간에 다루지는 않지만 방학 때 여러분이 직접 공부해보시기 바랍니다.

제1장 • 몰라도 일단 짜보는 객체지향 프로그램

이 장에서는 내용이 어렵기는 하지만 무조건 따라 하기 방식으로 객체지향의 맛을 보도록 하겠습니다. 무엇을 공부해야 할지 모르면 어떻게 공부해야 할지도 모르는 경우가 많습니다. 이 장에서는 객체지향 프로그램을 직접 작성해 봄으로써 무엇을 공부해야 할지를 미리 살펴보겠습니다.

제11장 • 객체지향 프로그래밍 실전 연습과 UML 모델링

객체지향 프로그래밍을 공부할 때 아쉬운 점의 하나는 스택과 같이 특정 기능을 구현하는 예제 코드는 많지만, OOP 원리를 쉽게 이해할 수 있는 예제 프로그램을 구하기가 어렵다는 것입니다. 이 장에서는 1장에서 작성한 프로그램을 하향식(top-down)으로 작성하면서 어떻게 역할 분담을 하고 객체들을 연결하는지에 대해 공부할 것입니다.

5. 수업 진행 방법

- 처음 2~3시간 수업 후 이후 남은 시간은 자기 주도 실습
- [Flip Learning] 또는 특정 내용에 대해 학생 스스로 선행학습 후 Q/A 및 자기주도 실습

6. 평가방법

중간고사 40% + 기말고사 40% + 과제물 10% + 출석 10%
(평가 항목 및 반영 비율은 변경될 수 있습니다.)

7. 주의사항

- 실습실에서 한번 정해진 자리는 바꾸지 않습니다(자리 이동 금지).
- 수업 시간 중 인터넷, 게임, 스마트폰 사용 금지 등
- **Visual Studio 2019 이상 설치되어 있는지 확인!**