IT COOKBOOK
한빛교재시리즈 141

꼭 필요한 내용만 간결하게 배우는 C의 핵심

원리를 알면 IT가 보인다

C로 시작하는 컴퓨터 프로그래밍

개정판

문호석, 손명호 지음

한빛아카데미
Hanbit Academy, Inc.

연습문제 해답은 제공하지 않습니다

>> 워밍업

이 책을 읽기 전에

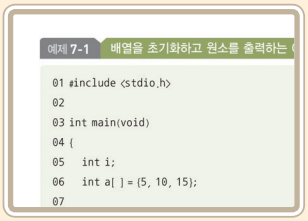
누구를 위한 책인가

프로그래밍을 전혀 해보지 않았거나, 도중에 포기한 이들을 위한 C 언어 입문서다. 학습자의 부담을 덜어 줄 수 있도록 꼭 알아야 할 내용만 간결하게 담되, 다양한 수준의 예제를 직접 프로그래밍하면서 복잡한 문법을 이해할 수 있도록 했다. 연습문제와 실전 프로그램을 통해 응용력까지 기를 수 있다.

선수 연계 과목

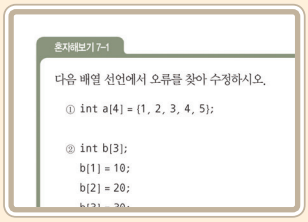
프로그래밍을 처음 해보는 이들이 대상이므로 특별한 선수 과목은 없다. 그렇지만 컴퓨터를 문서를 작성하고 인터넷을 검색하는 등 컴퓨터의 일반 기능을 사용할 수 있고, 컴퓨터와 친숙한 상태라면 좀더 쉽게 이 책의 내용을 따라올 수 있을 것이다.

이 책의 구성 요소



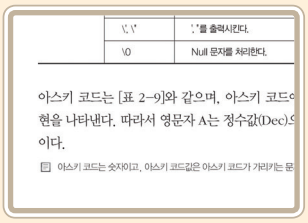
실습과 예제

개발 도구를 설치하는 실습과 기본 이론에 대한 이해를 돕는 예제 프로그램이다.



혼자해보기

예제 프로그램을 작성한 후에 비슷한 유형의 문제를 직접 풀어보며 응용력을 기를 수 있게 했다.



Tip

중요한 내용이나 관련 용어에 대한 부연 설명이다.

이 책의 뼈대만 빨리 보기

학습자가 부담을 덜 수 있도록 프로그래밍 입문자가 꼭 알아야되는 내용만으로 구성하였다. 준비 → 기본 → 심화 → 실전 순으로 진행한다.

❶ 준비 학습(1장~4장)

C 언어의 특징과 필요성을 이해시키고, 학습 동기를 부여한다. 먼저, 간단한 프로그램과 C 언어의 표준 입출력에 대해서 배우고 실습 방법을 학습한다. 다음으로 C의 자료형과 연산자에 대한 기본 개념을 배우면서 C 프로그래밍에 대한 감각을 익힌다.

❷ 기본 학습(5장~9장)

C 언어의 기본인 조건문과 반복문을 학습해 사용자 요구에 따른 실행 문장을 만드는 방법을 배우고, 배열을 통해 동일한 자료형을 동시에 선언하는 방법을 익힌다. 또, 함수를 통해 복잡한 문제를 기능별로 분리하고 단순화시키는 방법을 배움으로써, 중급 단계로 넘어가기 위한 기본을 다진다.

❸ 심화 학습(10장~12장)

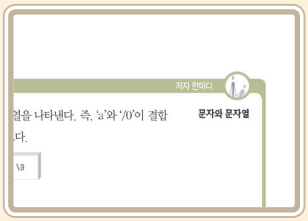
C 언어의 핵심인 포인터와 구조체에 대해 철저히 분석한다. 특히 앞서 배운 내용과 포인터의 연관성을 중점적으로 다루므로 좀더 수준 높은 프로그래밍을 배울 수 있다. 또한 파일 처리를 통해 프로그래밍 결과물을 만드는 재미도 맛볼 수 있다.

❹ 실전 예제(13장)

심화 학습 단계까지 학습한 많은 지식을 완성된 프로그램으로 담아내는 프로젝트를 진행한다. 데이터 정렬 방법과 영상처리 기법, 그리고 무기의 명중률 계산에 대해서 배움으로써, 알고리즘 및 현장에서 활용되는 프로그램도 배울 수 있다. 이 과정까지 끝나면 C 언어가 실제 업무에서 어떻게 활용되는지 알게 되고 프로그래밍에 대한 자신감까지 얻을 수 있을 것이다.

저자 한마디

저자가 현장에서 생각하고 발견한 유용한 팁, 함정을 미리 피해갈 수 있는 주의 사항, 참고로 알아두어야 할 사항을 별도로 정리했다.

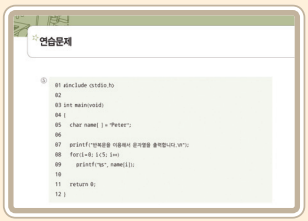


요약

장이 끝날 때마다 핵심 내용을 요약해서 정리한다. 해당 장에서 익힌 세분화된 지식을 여기서 전체적으로 조립하여 완성된 모습으로 볼 수 있다.

연습문제

장이 끝날 때마다 본문에서 익힌 내용을 연습문제를 해결하면서 점검한다. 이 문제를 풀면서 익힌 내용을 확인해보고 응용력을 기를 수 있다.



C 프로그래머 성장 전략 맵

프로그래밍을 잘하려면 어떻게 해야 하나요?

우선 프로그래밍에 대한 열정이 있어야 합니다. 열정과 끈질긴 승부 근성으로 문제를 풀기 위해 밤새워 고민하면, 문제를 해결했을 때 무한한 희열과 환희를 맛볼 수 있습니다. 그래야 프로그래밍에 재미가 붙고, 자신감이 생깁니다. 또한 다른 사람이 작성한 코드를 분석하고, 실제로 적용해보려고 끊임없이 노력해야 합니다. 스스로 이런 준비가 되어 있다고 판단한다면 프로그래머로서 자질이 충분한 사람입니다.

C는 C++를 학습하기 전에 반드시 먼저 학습해야 하는 과정인가요?

그렇습니다. C는 C++보다 먼저 학습해야 합니다. 일부에서는 C를 배우지 않고, C++를 먼저 배워도 괜찮다고 합니다. 하지만 그 경우도 C++ 강의 대부분을 C를 학습하는 데 보냅니다. C++는 C의 기본 기능에 객체 지향 개념이 추가된 것이기 때문입니다. 모든 것이 그렇듯 시간과 노력을 들여 단계를 밟아가야 진정한 전문가가 지녀야 할 기술력을 갖추게 됩니다.

이 책은 여기까지

1단계 : 기초 다지기

C 프로그래밍을 하려면 다음 항목을 제대로 익혀두어야 한다.

- 전산 개론
- 운영체제의 개념 및 운용 방법
- 각종 응용 프로그램의 사용법

목표 : 컴퓨터 관련 기본 지식을 바탕으로 응용 프로그램(특히 편집기) 이용에 익숙해져야 한다.
수준 : 학교에서 전산 개론이나 실용 전산 과정을 이수한다.

2단계 : 초보자 탈피하기

다음 내용으로 C 프로그래밍의 기본을 갖춘다.

- C 언어를 자유롭게 사용
- 배열의 개념 이해
- 함수를 이용한 모듈 설계

목표 : C의 언어적 특성을 이해하고 구조적인 프로그램을 제작할 수 있다.
수준 : 현장에서 기본은 한다.

4단계 : 전문가로 거듭나기

C 프로그래밍 관련 응용 분야를 이해하고, 다른 프로그래밍 언어 학습과 연계한다.

- 윈도우/멀티미디어 프로그래밍
- 네트워크 프로그래밍 • .NET/J2EE 프로그래밍
- 시스템 프로그래밍 • 게임 프로그래밍
- 컴포넌트 프로그래밍 • 모바일 프로그래밍
- 웹/데이터베이스 프로그래밍

목표 : 다양한 환경(언어와 플랫폼)에서의 실전 연습으로 고급 기술력 습득
수준 : 하고 싶은 일을 선택한다.

3단계 : 중급자로 거듭나기

고급 기법을 통해 좀더 빠르고 효율적인 실용 프로그램을 제작할 준비를 한다.

- 포인터와 배열의 관계 이해
- 구조체와 매크로에 대한 이해와 활용
- 알고리즘에 대한 이해

목표 : 기본적인 자료 구조 활용법과 프로젝트 수행
수준 : 현장에서 제 몫을 한다.

C 프로그래밍 관련 학습 로드맵

Chapter 05

선택문

＊학습 목표

- 프로그램의 흐름을 바꾸는 다양한 선택문을 알아본다.
- 대표적인 선택문인 if문과 switch문을 중점적으로 알아본다.

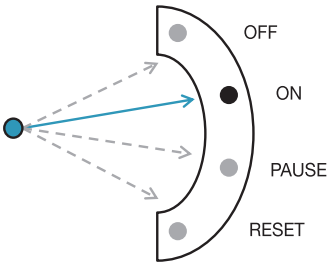
Section

03 switch~case문

1. switch~case문이란

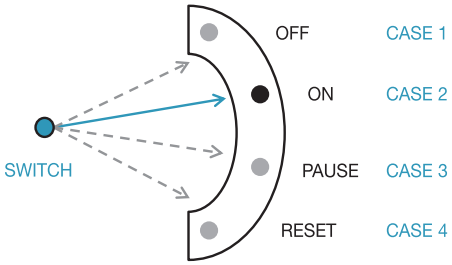
if문이 프로그램의 흐름을 2가지 상황으로 나누는데 반해 switch문은 조건식을 먼저 평가한 뒤 그 식의 값이 case 상수와 일치하는 쪽으로 분기하여 해당 명령문을 수행한다. 스위치를 움직이면 스위치가 가리키는 동작을 실행하는 기계 같은 역할을 하는 것이 바로 switch문이다.

[그림 5-3]은 특정 기계를 동작시킬 때 사용되는 모듈이다. 그림에서 보듯이 사용되는 기능에는 OFF, ON, PAUSE, RESET 4가지가 있다. 이 4가지 기능은 화살표가 가리킬 때 동작한다. 그림에서는 화살표가 ON을 가리키기 때문에 기계는 ON에 해당하는 동작을 수행한다.



[그림 5-3] 스위치 동작

[그림 5-4]는 [그림 5-3]에 나온 기계의 동작들을 프로그래밍적으로 표현한 것이다. [그림 5-4]에서 화살표를 스위치(SWITCH)라고 간주하고 각 기능을 CASE로 표현했다. 즉, 화살표는 4가지 CASE를 가리킬 수 있다. 예를 들어, 화살표가 ON을 가리키면 CASE 2에 해당하는 명령을 수행한다.



[그림 5-4] 스위치 동작과 switch문의 관계

switch문의 기본형식은 다음과 같다.

■ switch문 형식

```
switch(조건식)
{
   case 상수값 1 :
      명령문 블록 1
      break;
   case 상수값 2 :
      명령문 블록 2
      break;
      ...
   default :
      명령문 블록 n
      break;
}
```

switch문의 조건식 결과값과 case 상수값이 일치하는지 검사하여, 일치하는 값이 나오면 그 case문 아래의 명령문 블록을 수행한다. 그렇지만 여러 개의 case문 중에서 어느 하나라도 일치하는 값이 없으면 default 부분의 명령문 블록을 수행한다. default 부분은 switch~case 문을 사용할 때 반드시 있어야 하는 것은 아니지만 조건에 맞는 case가 없을 경우를 대비해 만들어주는 것이 좋은 프로그래밍 습관이다.

명령문 중간의 break문은 해당하는 case의 명령문을 빠져 나오는 데 필요하므로 빠뜨리지 않도록 주의한다. 만약 case 1에 break문이 없으면, case 2의 명령문까지 한꺼번에 처리되어 원하지 않는 결과가 나온다. 따라서 명령문 블록 1개가 끝나면 반드시 break문을 사용해야 한다.

break에 관한 자세한 내용은 152쪽에서 다룬다.

예제 5-5 **switch~case문 기초 예제** 프로그램 5-5.c

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05   int input;
06
07   printf("영어로 인사하는 법을 배우겠습니다.\n");
08   printf("아침 인사는 1번을 누르세요.\n");
09   printf("점심 인사는 2번을 누르세요.\n");
10   printf("저녁 인사는 3번을 누르세요.\n");
11   scanf("%d", &input);
12
13   switch(input){
14     case 1 :
15       printf("Good Morning!!\n");
16       break;
17     case 2 :
18       printf("Good Afternoon!!\n");
19       break;
20     case 3 :
21       printf("Good Night!!\n");
22       break;
23   }
24
25   return 0;
26 }
```

[1을 입력했을 때]

```
"C:₩Program Files₩Microsoft Visual Studio₩MyProjects₩...
영어로 인사하는 법을 배우겠습니다.
아침 인사는 1번을 누르세요.
점심 인사는 2번을 누르세요.
저녁 인사는 3번을 누르세요.
1
Good Morning!!
```

[2를 입력했을 때]

```
"C:₩Program Files₩Microsoft Visual Studio₩MyProjects₩...
영어로 인사하는 법을 배우겠습니다.
아침 인사는 1번을 누르세요.
점심 인사는 2번을 누르세요.
저녁 인사는 3번을 누르세요.
2
Good Afternoon!!
```

[3을 입력했을 때]

```
"C:₩Program Files₩Microsoft Visual Studio₩MyProjects₩...
영어로 인사하는 법을 배우겠습니다.
아침 인사는 1번을 누르세요.
점심 인사는 2번을 누르세요.
저녁 인사는 3번을 누르세요.
3
Good Night!!
```

[4를 입력했을 때]

```
"C:₩Program Files₩Microsoft Visual Studio₩MyProjects₩...
영어로 인사하는 법을 배우겠습니다.
아침 인사는 1번을 누르세요.
점심 인사는 2번을 누르세요.
저녁 인사는 3번을 누르세요.
4
```

[예제 5-5]에서 조건식은 input이다. 즉, input 값에 따라 3가지 경우로 실행되도록 프로그래밍했다. 입력한 값이 1~3이면 각 숫자에 따라 결과가 출력된다. 그러나 1~3이 아닌 다른 값을 입력하면 아무것도 실행되지 않는다. 오류는 발생하지 않지만 입력 범위를 벗어난 값을 처리하지 않았기 때문에 프로그램의 흐름이 매끄럽지 않다.

[예제 5-6]을 통해서 예외를 처리해보자.

예제 5-6 default를 사용하여 예외를 처리한 예제

프로그램 5-6.c

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05   int input;
06
07   printf("영어로 인사하는 법을 배우겠습니다.\n");
08   printf("아침 인사는 1번을 누르세요.\n");
09   printf("점심 인사는 2번을 누르세요.\n");
10   printf("저녁 인사는 3번을 누르세요.\n");
11   scanf("%d", &input);
12
13   switch(input){
14     case 1 :
15       printf("Good Morning!!\n");
16       break;
17     case 2 :
18       printf("Good Afternoon!!\n");
19       break;
20     case 3 :
21       printf("Good Night!!\n");
22       break;
23     default :
24       printf("잘못 입력했습니다. 1~3사이의 숫자를 입력하세요.\n");
25   }
26
27   return 0;
28 }
```

```
"C:₩Program Files₩Microsoft Visual Studio₩MyProjects₩ss₩Debug₩...
영어로 인사하는 법을 배우겠습니다.
아침 인사는 1번을 누르세요.
점심 인사는 2번을 누르세요.
저녁 인사는 3번을 누르세요.
4
잘못 입력하였습니다. 1~3사이의 숫자를 입력하세요.
```

default를 사용해 예외를 처리하면 프로그램의 흐름이 좀더 매끄럽다.

2. switch~case문의 조건식

switch~case문의 조건식에는 if문에서 사용했던 관계 연산자나 논리 연산자는 쓸 수 없고 반드시 수식이나 값을 넣어야 한다.

■ 조건식을 잘못 사용한 예

```
switch(input > 0)
{
   ...
   ...
   ...
}
```

■ 조건식을 올바르게 사용한 예

```
switch(input)
{
   ...
   ...
   ...
}
```

예제 5-7 switch문의 조건으로 수식을 사용한 예제

프로그램 5-7.c

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05   int score;
06
07   printf("당신의 점수를 입력하세요 :");
08   scanf("%d", &score);
09
10   switch(score/10)
11   {
12     case 10 :
13     case 9 : printf("점수는 %d점이고 성적은 %c입니다.\n", score, 'A'); break;
14
15     case 8 : printf("점수는 %d점이고 성적은 %c입니다.\n", score, 'B'); break;
16
17     case 7 : printf("점수는 %d점이고 성적은 %c입니다.\n", score, 'C'); break;
18
19     case 6 : printf("점수는 %d점이고 성적은 %c입니다.\n", score, 'D'); break;
20
21     default : printf("점수는 %d점이고 성적은 %c입니다.\n", score, 'F'); break;
22   }
23
24   return 0;
25 }
```

```
"C:₩Program Files₩Microsoft Visual Studio₩MyProjects₩ss₩Debug₩...
당신의 점수를 입력하세요 :93
점수는 93점이고 성적은 A입니다.
```

[예제 5-7]은 사용자에게 점수를 입력받아 학점을 계산하는 프로그램이다. 학점은 아래와 같이 계산한다.

- 90점 이상 : A학점
- 80점 이상 90점 미만 : B학점
- 70점 이상 80점 미만 : C학점
- 60점 이상 70점 미만 : D학점
- 60점 미만 : F학점

[예제 5-7]은 조건식으로 입력받은 점수를 10으로 나누는 수식을 썼고, case문의 상수값은 나눈 몫을 이용했다. case 10에는 break가 없으므로 case 9와 동일하게 처리해 100점인 경우 A로 평가한다. 그리고 60점 미만의 점수는 default로 처리했다.

3. break문의 사용

break문은 제어문을 강제로 종료시킨다. 즉 정상적인 종료 조건이 아니더라도 break문을 만나면 제어문은 종료된다.

switch~case문에서는 break문을 사용하여 선택된 case에 해당하는 것만 실행하는데, break문의 적절한 사용은 중요하다. [예제 5-8]은 break문을 적절하게 사용하지 못해서 발생하는 문제를 다룬다.

예제 5-8 **break문을 잘못 사용한 예제** 프로그램 5-8.c

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05   int input;
06
07   printf("영어로 인사하는 법을 배우겠습니다.\n");
08   printf("아침 인사는 1번을 누르세요.\n");
09   printf("점심 인사는 2번을 누르세요.\n");
10   printf("저녁 인사는 3번을 누르세요.\n");
11   scanf("%d", &input);
12
13   switch(input){
14     case 1 :
15       printf("Good Morning!!\n");
16     case 2 :
17       printf("Good Afternoon!!\n");
18     case 3 :
19       printf("Good Night!!\n");
20       break;
21     default :
22       printf("잘못 입력했습니다. 1~3사이의 숫자를 입력하세요.\n");
23   }
24
25   return 0;
26 }
```

```
"C:₩Program Files₩Microsoft Visual Studio₩MyProjects₩ss₩Debug₩...

영어로 인사하는 법을 배우겠습니다.
아침 인사는 1번을 누르세요.
점심 인사는 2번을 누르세요.
저녁 인사는 3번을 누르세요.
1
Good Morning!!
Good Afternoon!!
Good Night!!
```

[예제 5-8]에서 1이 입력되면 아침, 점심, 저녁 인사와 default의 내용이 전부 출력된다. case문에 break문이 없기 때문에 case 1에서 종료되지 않고 아래로 계속 진행된 것이다. 위 예제에서 2가 입력되면 점심, 저녁 인사와 default의 내용이 출력된다. 즉, case 2 아래의 모든 문장이 실행되는 것이다. case별로 하는 일이 각각 다를 경우에는 break문을 꼭 사용해야 한다.

한편 break문을 적절하게 사용하면 효율적으로 프로그래밍할 수 있다. 예를 들어, 평점에 따라 장학금과 해외연수, 겨울특강 수강자격을 주는 학교 포상이 있다고 하자. 만약, 평점이 4.4이상이면 3가지 특혜를 다 받고, 4.3이상 4.4미만이면 장학금과 겨울특강 수강자격을, 4.2이상 4.3미만이면 겨울특강 수강자격만 받는 프로그램을 switch~case문을 이용해서 처리할 때 break문을 이용하면 좋다.

다음 예제를 통해 break문을 적절하게 사용한 경우와 그렇지 않은 경우를 비교해보자.

■ break문을 적절하게 사용하지 않은 프로그램

```
// input이 1일 경우 : 4.2 <= 평점 < 4.3
// input이 2일 경우 : 4.3 <= 평점 < 4.4
// input이 3일 경우 : 4.4 <= 평점

switch(input)
{
   case 1 :
      printf("겨울특강 수강자격을 받습니다.!!\n");
      break;
   case 2 :
      printf("겨울특강 수강자격을 받습니다.!!\n");
      printf("장학금을 받습니다.\n");
      break;
   case 3 :
      printf("겨울특강 수강자격을 받습니다.!!\n");
      printf("장학금을 받습니다.\n");
      printf("해외연수 기회가 주어집니다.\n");
      break;
   default :
      printf("잘못 입력했습니다. 1~3사이의 숫자를 입력하세요.\n");
}
```

break문을 적절히 사용하지 않으면 위와 같이 같은 내용을 여러 번 작성해야 한다. 따라서 다음과 같이 break문을 적절하게 사용해야 좀더 효율적으로 프로그래밍할 수 있다.

■ **break문을 적절하게 사용한 프로그램**

```
// input이 1일 경우 : 4.2 <= 평점 < 4.3
// input이 2일 경우 : 4.3 <= 평점 < 4.4
// input이 3일 경우 : 4.4 <= 평점

switch(input)
{
   case 3 :
      printf("해외연수 기회가 주어집니다.\n");
   case 2 :
      printf("장학금을 받습니다.\n");
   case 1 :
      printf("겨울특강 수강자격을 받습니다.!!\n");
      break;
   default :
      printf("잘못 입력했습니다. 1~3사이의 숫자를 입력하세요.\n");
}
```

위 예에서 input이 3인 경우에는 받을 수 있는 3가지 혜택이 뜬다. 사용자에게 숫자를 입력받는 부분을 포함하여 소스코드를 완성하면 [예제 5-9]와 같다.

예제 5-9 break문을 적절하게 사용한 예제

프로그램 5-9.c

```
01 #include <stdio.h>
02 int main(void)
03 {
04    int input;
05
06    printf("평점이 4.2이상 4.3미만이면 1번을 누르세요.\n");
07    printf("평점이 4.3이상 4.4미만이면 2번을 누르세요.\n");
08    printf("평점이 4.4이상이면 3번을 누르세요.\n");
09
10    scanf("%d", &input);
11
12    switch(input){
13      case 3 :
14        printf("해외연수 기회가 주어집니다.\n");
15      case 2 :
16        printf("장학금을 받습니다.\n");
17      case 1 :
18        printf("겨울특강 수강자격을 받습니다.!!\n");
19        break;
20      default :
21        printf("잘못 입력했습니다. 1~3사이의 숫자를 입력하세요.\n");
22    }
23
24    return 0;
25 }
```

```
"C:₩Program Files₩Microsoft Visual Studio₩MyProjects₩ss₩Debug₩...
평점이 4.2이상 4.3미만이면 1번을 누르세요.
평점이 4.3이상 4.4미만이면 2번을 누르세요.
평점이 4.4이상이면 3번을 누르세요.
3
해외연수 기회가 주어집니다.
장학금을 받습니다.
겨울특강 수강자격을 받습니다.
```

원리를 알면 IT가 맛있다

IT COOKBOOK C로 시작하는 컴퓨터 프로그래밍 개정판

어렵다는 오해와 편견 대신
재미와 자신감을 주는 C언어 입문서

▸ 누구를 위한 책인가?

프로그래밍을 전혀 해보지 않았거나, 도중에 포기한 이들을 위한 C 언어 입문서다. 학습자의 부담을 덜어줄 수 있도록 꼭 알아야 할 내용만 간결하게 담되, 다양한 수준의 예제를 직접 프로그래밍하면서 복잡한 문법을 이해할 수 있도록 했다. 연습문제와 실전 프로그램을 통해 응용력까지 기를 수 있다.

▸ 개정판에서 달라진 부분은 무엇인가?

비주얼 C++ 6.0을 기반으로 하면서 2008 이상에서도 실습 가능하도록 추가 자료 제공, 기존 내용 수정/보완, 연습문제와 실전예제 추가

▸ 무엇을 다루는가?

준비	1~4장	• C언어의 시작과 구조 • 기본 자료형과 형변환	• C 프로그램 개발 환경 • 표준 입출력	• 디버깅 • 연산자
기본	5~9장	• 제어문의 종류 • for문 • 배열	• if문 • while문 • 함수	• switch~case문 • 기타 제어문 • 변수 영역
심화	10~12장	• 포인터 • 구조체 • 파일 입출력	• 포인터와 배열 • 구조체 접근과 사용 • 파일 입출력 함수	• 포인터와 문자열 • 열거형 • 매크로
실전	13장	• 데이터 정렬 프로그램	• 간단한 영상처리 프로그램	• 명중률 구하기 프로그램

프로그래밍 / C언어

93000

9 788998 756925

정가 18,000원

ISBN 978-89-98756-92-5