

Section

2 C 프로그램 개발 환경

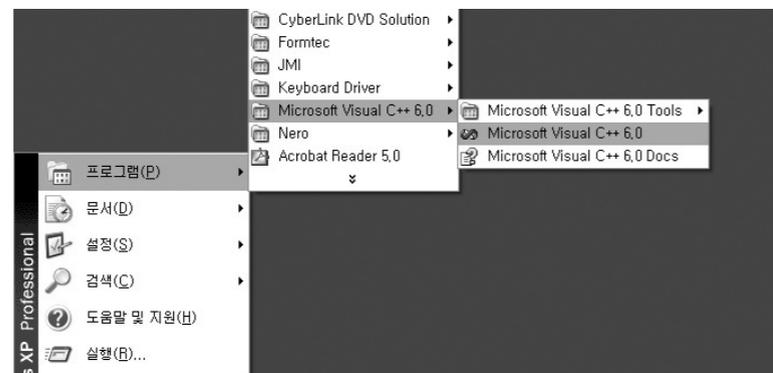
1 Visual C++를 이용한 프로그램 작성 방법

프로그램의 개발 과정은 편집-컴파일/링크-실행 단계로 구분된다. 이러한 과정을 하나의 프로그램 내에서 제공하는 것을 통합개발환경(IDE, Integrated Development Environment)이라 한다. 도스에서는 Turbo-C나 Borland-C가 통합개발환경을 제공하며, 윈도우에서는 Visual C++ 또는 C-Builder가 통합개발환경을 제공한다.

이 책에서는 Visual C++ 통합개발환경을 이용한다. Visual C++를 이용해서 “저는 경영과 감경영입니다”를 도스창에 출력하는 프로그램을 단계별로 작성해보자.

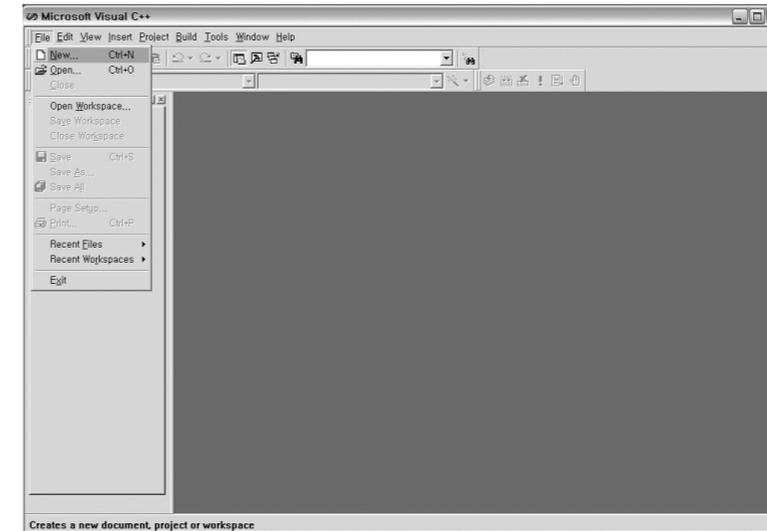
실습 1-1 도스창에 텍스트를 출력하는 프로그램 작성하기

1 Microsoft Visual Studio에 있는 Microsoft Visual C++ 6.0을 선택한다.



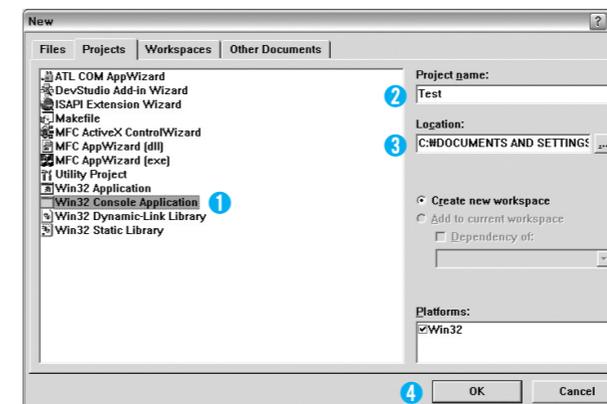
[그림 1-3] Visual C++ 6.0 프로그램 선택

2 프로그램 편집창에서 [File]-[New] 메뉴를 클릭한다.



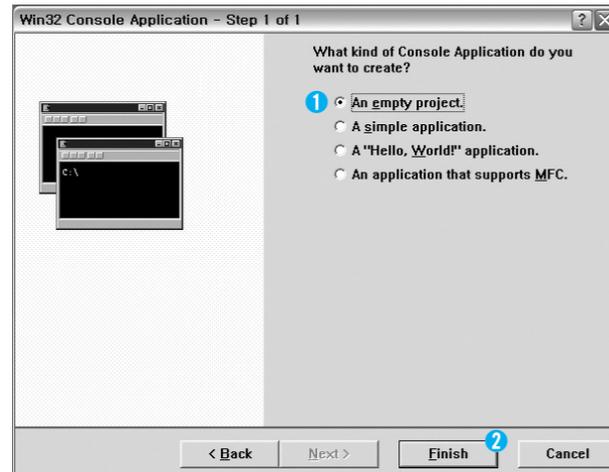
[그림 1-4] [File]-[New] 메뉴 선택

3 [그림 1-5]와 같은 [New] 대화상자가 나타나면 ① Win32 Console Application을 선택한다. ② Project name에 프로젝트명 'Test'를 입력한다. ③ Location 텍스트 상자에 파일 만들 위치를 선택하고 ④ <OK> 버튼을 클릭한다.



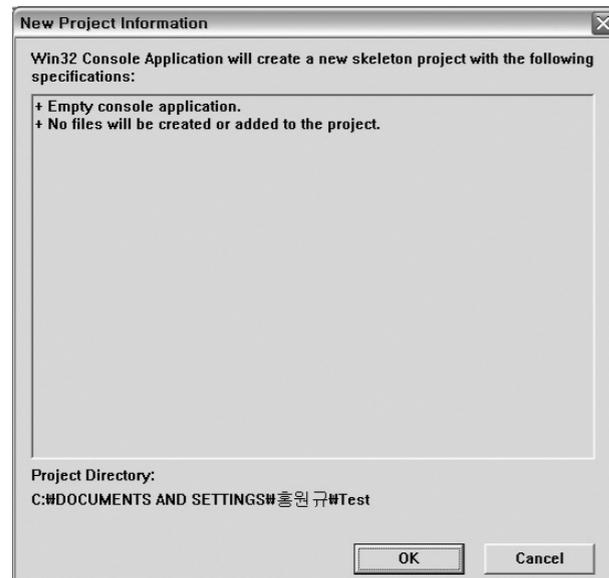
[그림 1-5] Win32 Console Application 선택

- 4 [Win32 Console Application] 대화상자에서 ① An empty project를 선택하고 ② <Finish> 버튼을 클릭한다.



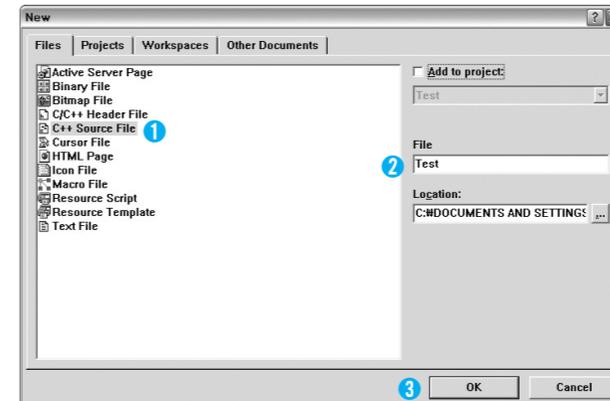
[그림 1-6] An empty project 선택

- 5 [New Project Information] 대화상자에서 <OK> 버튼을 클릭한다.



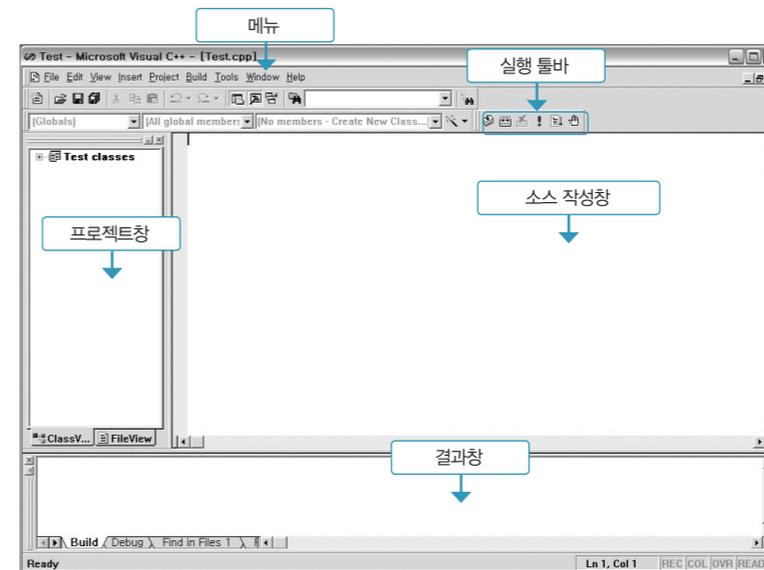
[그림 1-7] Project 만들기 성공

- 6 다시 프로젝트 편집창 메뉴에서 [File]-[New] 메뉴를 클릭하면 [그림 1-8]과 같은 [New] 대화상자가 나타난다. [Files] 탭에서 ① C++ Source File을 선택하고 ② 오른쪽의 File에 새로 작성할 파일명 'Test'를 입력한다. 다 입력했으면 ③ <OK> 버튼을 클릭한다.



[그림 1-8] 소스 파일 만들기

- 7 [그림 1-9]와 같이 프로그램 편집창이 열린다. 편집창의 명칭은 다음과 같다.



[그림 1-9] 프로그램 편집창 설명

- 소스 작성창: C 언어 문법에 맞추어 사용자가 프로그램을 작성하는 창이다.
- 실행 툴바: 사용자가 작성한 소스코드를 컴파일, 빌드(링크), 실행시킬 때 사용하는 창이다. [그림 1-10]과 같이 툴바를 사용한다.



[그림 1-10] 실행 툴바 설명

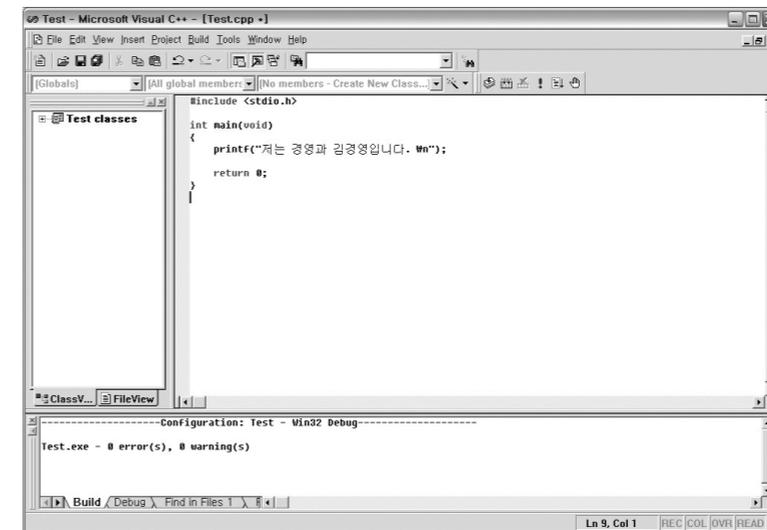
- 프로젝트창: 사용자 만든 프로젝트가 표시된다.
- 결과창: 소스파일을 컴파일하고 빌드한 결과가 나타난다. 소스파일을 컴파일한 .obj 파일의 오류 유무와 빌드한 .exe 파일의 오류 유무가 표시된다.

8 이제 프로그램을 실행시키기 위해 다음과 같이 소스 작성창에 프로그램 소스코드를 입력한다. 앞의 숫자는 독자가 알기 쉽게 줄번호를 매긴 것이다. 이 숫자는 소스 작성창에 입력하지 않는다.

```

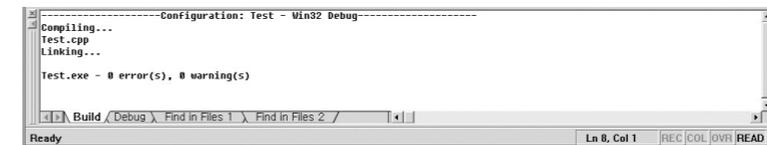
01 #include <stdio.h>
02
03 int main(void)
04 {
05     printf("저는 경영과 김경영입니다. \n");
06     return 0;
07 }
    
```

9 소스코드를 입력한 후 실행 툴바를 이용하여 컴파일한 후 빌드한다. 편집창의 아래 부분에 컴파일 결과가 표시되는데 error가 0이면 성공한 것이다. 만약 error가 발생하면 프로그램을 수정한 후 다시 컴파일해야 한다.



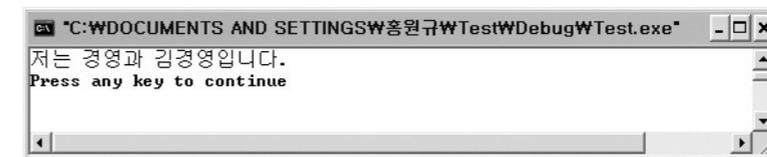
[그림 1-11] 소스 작성과 실행

10 컴파일과 빌드를 하면 [그림 1-12]와 같이 결과창에 실행 결과가 표시된다.



[그림 1-12] 결과창에 실행 결과 표시

11 실행 툴바의 실행 아이콘을 눌러 프로그램을 실행시킨다. 그러면 [그림 1-13]과 같이 실행 결과가 나타난다. 만약 결과 화면이 빨리 꺼진다면, **Ctrl** + **F5**로 실행시킨다.



[그림 1-13] 파일 실행 결과

Visual C++를 이용해서 간단한 C 프로그램을 작성해보았다. 아직 C 언어를 본격적으로 배우지는 않았지만 연습하는 의미에서 이 프로그램의 내용을 하나씩 살펴보고 넘어가자. 혹시 내용이 이해되지 않아도 앞으로 어떤 내용을 배우는지 가볍게 읽어 보자. 당연한 얘기겠지만 해당 내용은 뒤에서 자세하게 다룬다.

■ #include <stdio.h>

- 전처리기 부분이다. 전처리기 지시자인 #을 사용해서 include 명령을 실행한 것이다.
- include 명령어 다음에는 반드시 스페이스로 1칸을 띄우고 헤더파일명을 입력한다.
- 헤더파일은 확장자가 .h인 파일이다.
- 시스템(컴파일러를 가리킴)에 있는 헤더파일을 사용할 경우에는 아래와 같은 형태로 사용한다([실습1-1]에서는 printf() 함수를 사용하기 위해서 아래와 같이 작성했다). 여기서 () 나 {}를 사용해서는 안 되고 반드시 <>를 사용해야 한다는 점에 유의한다.

```
#include <헤더파일명> // 시스템 헤더파일
```

- 시스템에 있는 헤더파일이 아니라 사용자가 만든 헤더파일을 사용할 경우에는 <>를 사용하지 않고 아래와 같이 큰 따옴표를 사용한다. 사용자가 만든 헤더파일에 관한 자세한 내용은 8장에서 다룬다.

```
#include "헤더파일명" // 사용자가 만든 헤더파일
```

■ int main(void)

- main 다음에는 ()를 사용한다. () 안에 있는 void를 기록하지 않으면 경고가 발생할 수도 있으므로 작성하는 습관을 들이자.
- main 함수 아래에는 {}를 사용해야 한다.

■ printf("저는 경영과 김경영입니다. %n")

- printf() 함수는 괄호 안의 큰 따옴표 사이에 있는 내용을 도스 화면에 출력하는 함수다.
- 화면에 출력되는 내용은 괄호 안의 큰 따옴표 사이에 있는 값이다.

- \n은 도스 화면에 출력될 때 한 줄을 띄우라는 명령이다. 즉 **Enter**와 동일한 역할을 한다. `는 키보드에서 **₩**를 누르면 된다.

■ return 0

- 숫자 0을 반환하라는 의미다.
- int main() 함수에서 앞의 int는 정수형을 나타내는 자료형인데, main 함수가 정수값을 반환한다는 의미다. 이에 대한 자세한 사항은 8장에서 다룬다. 정수형 값을 반환하면 된다는 의미로 0 이외의 다른 정수값을 사용해도 된다.

Section

3 간단한 프로그램 오류수정과 환경설정

1 간단한 프로그램 오류수정 방법

프로그램을 작성할 때 문법에 맞지 않거나 또는 문법적인 문제는 없으나 논리적으로 맞지 않는 소스코드를 작성하면 오류가 발생한다. 오류는 문법적인 오류와 논리적인 오류로 나눌 수 있는데, 문법적인 오류는 쉽게 찾을 수 있으나 논리적인 오류는 찾기가 어렵다. 예를 들어 +를 해야 할 부분에 -를 한 것과 같은 오류는 프로그램 소스코드를 한 줄씩 실행시켜 보기 전에는 찾기 어렵다. 프로그램에 문법적인 오류나 논리적인 오류를 찾아 수정하는 것을 디버깅(Debugging)이라고 한다.

[실습1-1]의 내용을 이용해서 디버깅을 통해 오류를 제거하는 방법을 알아보자.

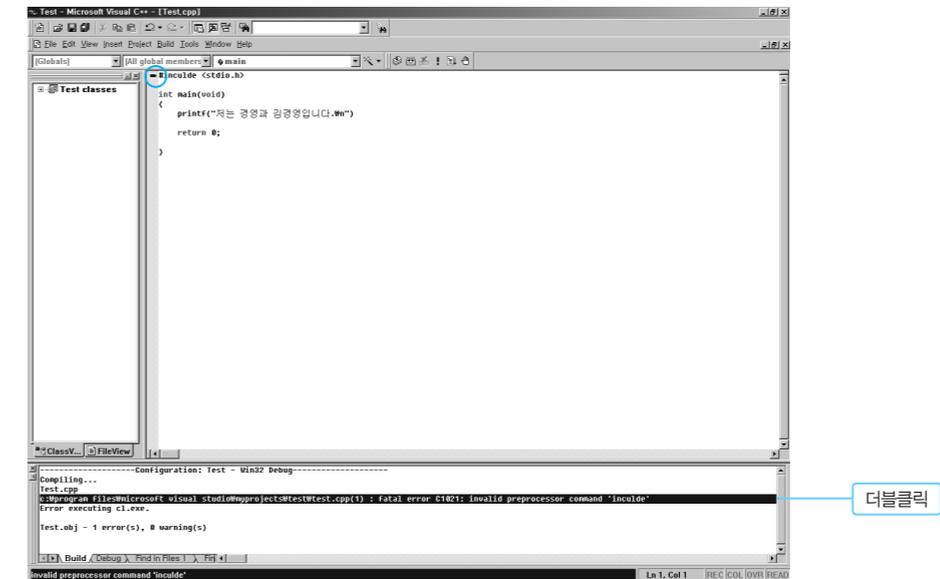
```
#include <stdio.h>                // include의 철자가 틀리다.
int main(void)
{
    printf("저는 경영과 김경영입니다. \n") // 마지막에 세미콜론이 없다.
    return 0;
}
```

위 소스코드는 주석에서 밝힌 것처럼 문법적인 오류가 2개 있다. 이 소스코드를 컴파일하면 [그림 1-14]와 같은 메시지가 결과창에 나타난다.



[그림 1-14] 결과창에 오류의 종류가 나타남

디버깅 메시지를 보면 invalid preprocessor command 'includ'이라고 기록되어 있다. 즉 includ는 유효하지 않은 전처리 명령어라는 얘기다. 이러한 오류 메시지가 발생한 소스코드로 이동하고 싶으면 결과창의 오류 메시지를 더블클릭한다.



[그림 1-15] 소스코드에서 오류의 위치를 알려줌

[그림 1-15]와 같이 화살표를 통해 오류가 발생한 소스코드의 위치를 알 수 있다. 물론, 위와 같이 소스코드가 간단하면 이 방법을 이용하지 않아도 알 수 있으나 소스코드의 양이 많다면 위와 같은 방법을 이용하여 찾아야 한다.

includ를 include로 수정한 후, 프로그램을 다시 컴파일해보면 [그림 1-16]과 같이 결과창에서 오류가 또 있음을 알려준다.



[그림 1-16] 또 다른 오류에 대한 알림

디버깅 메시지를 보면 missing ';' before 'return'이라고 기록되어 있다. 즉, return 명령어 앞에 ;이 없다는 뜻이다. 이 부분을 수정한 후 컴파일하면 이제 오류가 없다는 메시지가 나온다.



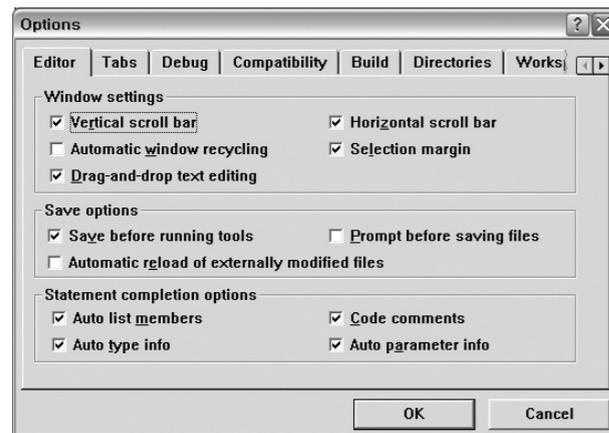
[그림 1-17] 오류를 제거한 후의 결과

2 Visual C++ 환경설정

여기서는 통합개발환경으로 이용되는 Visual C++에서 프로그램 작성 환경을 변경하는 방법을 간단히 소개한다. 이 부분은 C 언어를 공부하는데 직접적인 관련은 없으므로 참고 내용으로 보자.

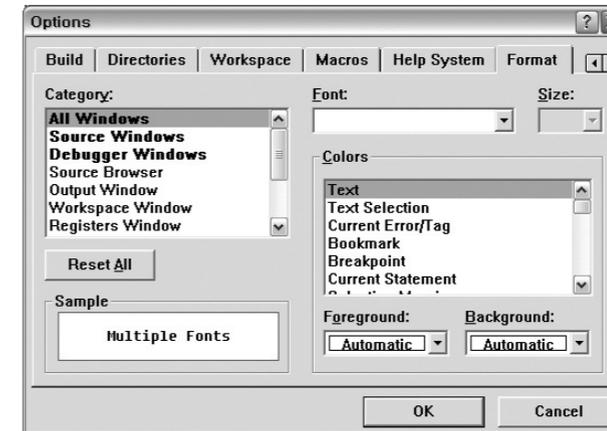
실습 1-2 Visual C++에서 환경 설정하기

- 1 프로그램 편집창에서 [Tools]-[Options] 메뉴를 클릭한다. 그러면 [그림 1-18]과 같은 대화상자가 나타난다.



[그림 1-18] [Tools]-[Options] 메뉴 선택

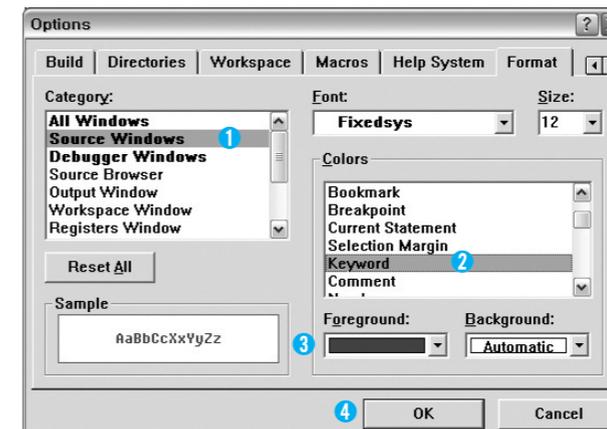
- 2 [Options] 대화상자에서 화살표를 이동해 [Format] 탭을 클릭하면 [그림 1-19]과 같은 대화상자가 나온다. 여기서 Visual C++ 통합개발환경에서 사용하는 글자의 형태를 바꿀 수 있다.



[그림 1-19] [Format] 탭 화면

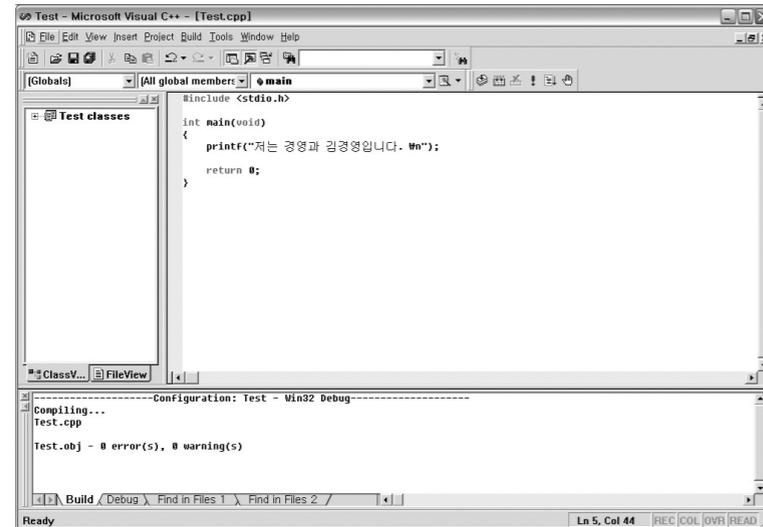
- 3 [Format] 탭 왼쪽의 Category 부분을 보면 Source Windows(소스 입력창), Debugger Windows(결과창) 등 통합개발환경 내에서 지정할 수 있는 범주 목록이 보인다. 이것을 이용해서 필요한 부분의 글자(Foreground) 및 배경(Background) 색과 글꼴 등을 바꿀 수 있다. 예로 소스 입력창에 있는 내용을 수정해보자.

① Category에서 Source Windows를 선택한다. ② 오른쪽의 Colors의 리스트 상자를 아래로 내려 Keyword를 선택한다. Keyword는 프로그램에서 미리 명령어로 만들어져 있는 단어로 앞의 예제에서 int같은 것이다. ③ Keyword는 기본적으로 파란색인데 이것을 빨간색으로 바꾸어보자. ④ <OK> 버튼을 클릭하여 대화상자를 닫는다.



[그림 1-20] Keyword 글자색 수정

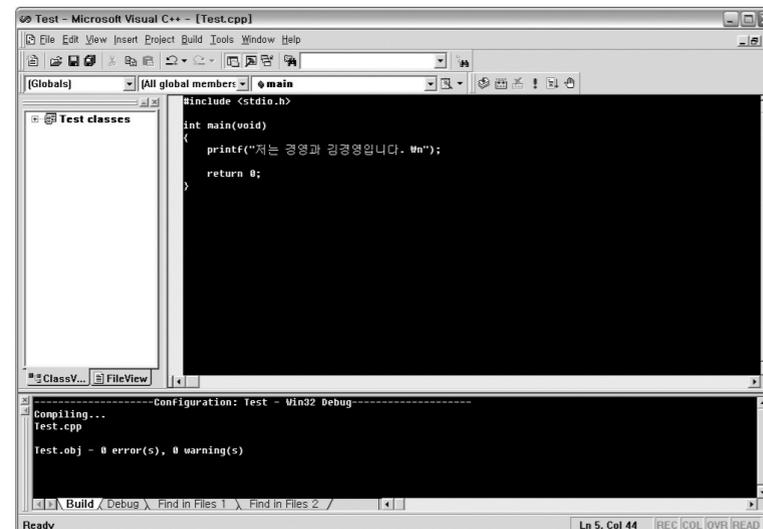
4 소스 입력창을 확인해 보면, 키워드의 색이 빨간색으로 바뀌었다.



[그림 1-21] keyword 색을 빨간색으로 수정한 결과

이 책에서는 색을 구분할 수 없으니 직접 실행해서 확인해 보자.

5 같은 방법으로 keyword와 text의 글자를 흰색으로, 배경을 검은색으로 바꾸면 [그림 1-22]와 같이 된다.

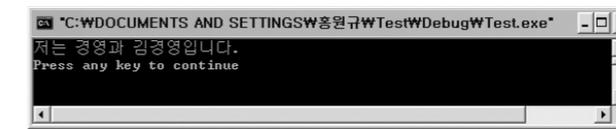


[그림 1-22] 글자색과 배경색을 바꾼 결과

이와 같이 [Options] 메뉴를 이용하면 사용자가 원하는 형태로 통합개발환경을 바꿀 수 있다.

3. 도스창 환경설정

프로그램을 실행시켰을 때 나타나는 도스창도 사용자가 원하는 형태로 바꿀 수 있다. 기본적으로 도스창은 [그림 1-23]과 같이 검은색 바탕에 흰색 글자로 출력된다.

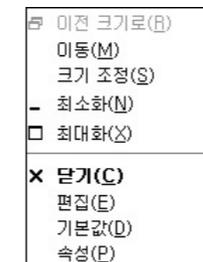


[그림 1-23] 기본적인 도스창의 모습

이제 기본적인 출력 형태를 여러분들이 원하는 형태로 바꿔보자.

실습 1-3 도스창 환경 설정하기

1 도스창 제목표시줄 위에 마우스를 올려놓고 마우스 오른쪽 버튼을 누르면 아래와 같은 메뉴가 나타난다.



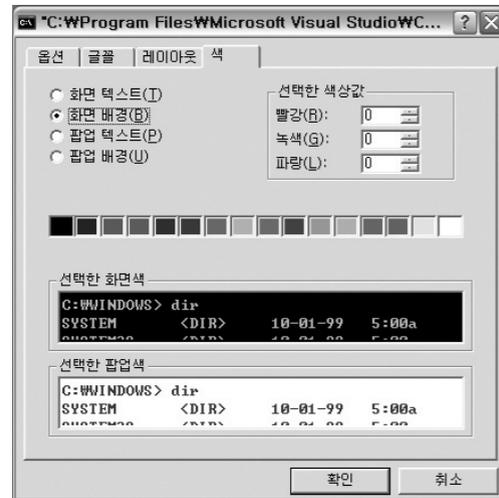
[그림 1-24] 도스창을 수정하기 위한 메뉴

2 속성을 클릭하면 [그림 1-25]와 같은 화면이 나타난다. 라디오 버튼에 화면 텍스트와 화면 배경이 있는데, 이것을 각각 클릭하여 화면 텍스트는 검은색으로 선택하고 화면 배경은 흰색으로 선택한 후 <확인>을 클릭한다.



★ 요약

Chapter 01



[그림 1-25] 도스창의 색을 바꾸기 위한 화면

- 3 변경한 도스창은 [그림 1-26]과 같다. 화면 배경은 흰색이고 화면 텍스트는 검은색으로 바뀐 것을 볼 수 있다.



[그림 1-26] 화면 배경과 텍스트를 색을 바꾼 결과

1 컴퓨터 시스템 구조

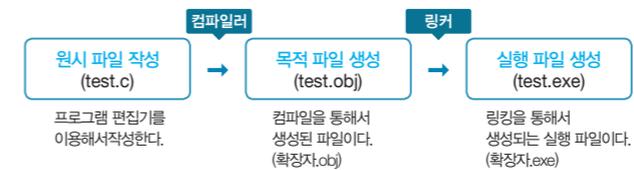
컴퓨터는 하드웨어와 소프트웨어로 구성된다. 하드웨어는 전자회로와 물리적인 장치로 이루어진 것을 말하고, 소프트웨어는 하드웨어를 움직이도록 하는 명령의 집합을 말한다.

2 소프트웨어의 분류

소프트웨어는 시스템 소프트웨어와 응용 소프트웨어로 나뉜다. 시스템 소프트웨어는 운영체제와 같이 컴퓨터 시스템 전반을 효율적으로 운영하고 제어하기 위해 만든 프로그램이며, 응용 소프트웨어는 한글과 워드, 엑셀 등과 같이 특정한 목적을 수행하도록 만든 프로그램이다.

3 프로그램 실행 절차

먼저 편집기를 통해서 원시 파일을 작성하고 작성된 원시 파일은 컴파일을 통해 목적 파일로 바뀐다. 컴파일을 통해 만들어진 목적 파일은 링커를 통해 시스템 라이브러리와 연결되며, 그 결과 실행 파일이 만들어진다.



4 프로그램 환경설정

프로그램 편집기의 배경과 실행된 도스창의 배경은 사용자가 원하는 형태로 바꿀 수 있다. 프로그램 편집기는 [Tools]-[Options] 메뉴를 클릭하여 바꿀 수 있고, 도스창은 도스창 제목 표시줄에서 마우스 오른쪽 버튼을 누른 후 [속성]을 선택하면 바꿀 수 있다.