

제1장

C# 프로그래밍을 위한 OOP 기본

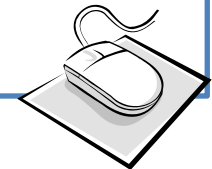
제주대학교 컴퓨터공학과
변영철 교수

1. 아주 간단한 C 프로그램

- Console1 프로젝트 생성
- 코드 작성
- 컴파일 및 실행

```
#include <stdio.h>

void main()
{
    printf("Hello, World!\n");
}
```



2. 조금 복잡한 C 프로그램

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    int iX;
```

```
    int iY;
```

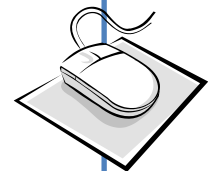
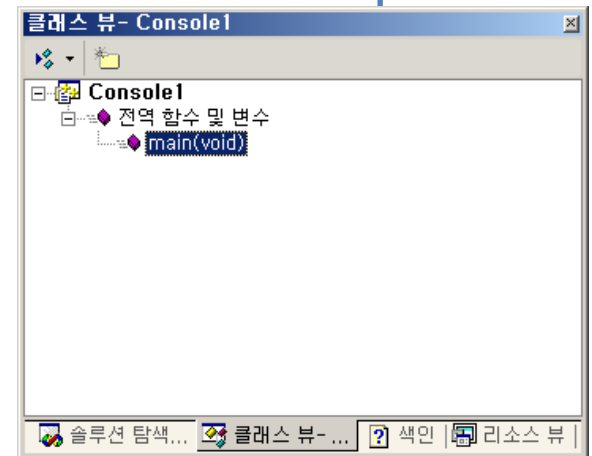
```
    iX = 2;
```

```
    iY = 3;
```

```
    int iResult = iX + iY;
```

```
    printf("두 개의 값을 더한 결과: %d\n", iResult);
```

```
}
```



2. 조금 복잡한 C 프로그램

- 변수의 종류
 - 지역(local) 변수 : 함수 안에서 만든 변수
 - 전역(global) 변수 : 함수 밖에서 만든(정의한) 변수
- 사용 범위(scope)
 - 지역 변수 : 변수를 정의한 함수 안에서만 접근
 - 전역 변수 : 모든 함수에서 접근
- 생명주기(life time)
 - 지역 변수 : 함수가 실행될 때 만들어지고 함수가 끝날 때 사라짐
 - 전역 변수 : 프로그램이 실행될 때 만들어지고, 프로그램이 종료될 때 사라짐

3. 함수를 이용한 프로그램

- 수 십 줄로 코드가 작성된다면?
 - 코드가 복잡하게 보이고 읽기(분석하기) 힘들.
 - 이를 해결하는 방법 → 함수

3. 함수를 이용한 프로그램

- 추상화의 의미
 - 묶어서, 간단히 표현하는 것
- 코드를 묶어서 간단히 표현 : 코드 추상화

```
#include <stdio.h>

void main()
{
    int iX;
    int iY;

    iX = 2;
    iY = 3;

    int iResult = iX + iY;

    printf("두 개의 값을 더한 결과: %d\n", iResult);
}
```

3. 함수를 이용한 프로그램

```
#include <stdio.h>
```

```
int iX;  
int iY;
```

```
void Assign()  
{  
    iX = 2;  
    iY = 3;  
}
```

```
int Add()  
{  
    return iX + iY;  
}
```

질문1. iX, iY를 이용하는 함수는?

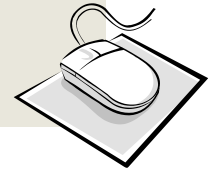
질문2. Assign 함수를 호출하는 함수는?

```
void main() {  
    int iResult;
```

```
    Assign();
```

```
    iResult = Add();
```

```
    printf("두 개의 값을 더한 결과: %d\n", iResult);  
}
```



3. 함수를 이용한 프로그램

```
#include <stdio.h>
```

```
int iX;  
int iY;
```

```
void Assign(int x, int y)  
{  
    iX = x;  
    iY = y;  
}
```

```
int Add()  
{  
    return iX + iY;  
}
```

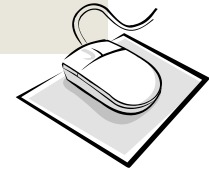
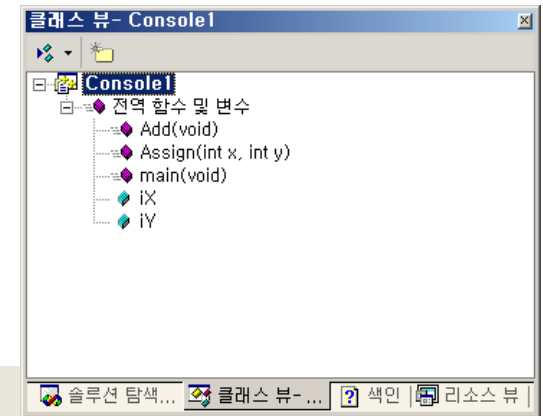
```
void main() {  
    int iResult;
```

값을 지정할 수 있도록 (파라미터)

```
    Assign(2, 3);
```

```
    iResult = Add();
```

```
    printf("두 개의 값을 더한 결과: %d\n", iResult);  
}
```



4. 100개의 변수와 1000개의 함수?

- 많은 변수와 함수들이 뒤죽박죽 섞여 있으면
 - 어떤 함수가 어떤 변수를 이용하는지 쉽게 알 수 없다!
 - 어떤 함수가 어떤 함수를 호출하는지 쉽게 알 수 없다!
 - 서로 관련없는 함수와 변수들이 기준 없이 뒤섞여 있어서 재사용시 어디까지 사용해야 하는지 쉽게 알 수 없다!
 - 변수를 초기화하는 함수는 꼭 한번 실행해야 하는데, 이 함수를 알아보기 어렵다!

4. 100개의 변수와 1000개의 함수?

- 따라서,
 - 오류가 발생할 경우 오류 수정이 어렵다.
 - 잘못된 접근을 방지할 수 있는 장치가 없다.
 - 코드 재사용이 어렵다.
 - 결국 **소프트웨어 위기**가 발생할 수 있다.

4. 100개의 변수와 1000개의 함수

- 이를 해결하려면?
 - 관련된 **변수**(데이터)와 함수 **코드**를 한 곳으로 모아 묶어야 함.
 - **코드 변수**(데이터) 추상화
 - 데이터 추상화