



## Target: Game Over Game

- “Game Over !” 를 화면에 출력하는 게임
  - 단순한 방법



- 약간 화려한 방법 (ASCII Art 사용)



2

# 1장 학습 목표



- 프로그래밍과 프로그래밍 언어를 이해한다.
- 절차적, 구조화, 객체지향 프로그래밍의 차이를 이해한다.
- 통합 프로그래밍 개발환경을 활용하는 능력을 기른다.
- 입출력문을 사용할 수 있는 능력을 기른다.
- 이스케이프 시퀀스의 의미를 이해한다.
- 아스키 아트를 이해하고 활용할 수 있다.
- 파일, 폴더, 확장자, 솔루션, 프로젝트, 소스 파일 등을 이해한다.

3

## 1.1 C++ 개요



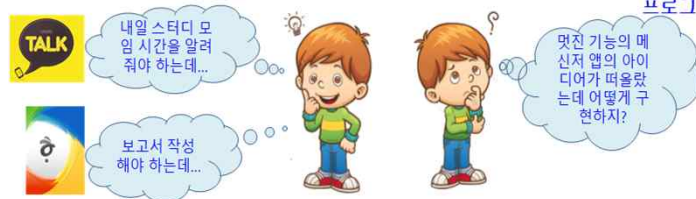
- 프로그래밍이란?
- C++의 탄생
- C++의 특징

4

## 프로그래밍이란?



- 컴퓨터에서의 문제 해결
  - 도구가 있다면 → 문서 작성 툴, CAD, 메신저 프로그램 등 사용



- 도구가 없다면 → **프로그래밍** 필요

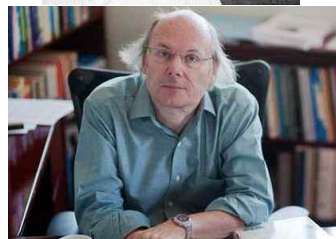
- **프로그래밍 언어**
  - 컴퓨터에게 원하는 절차를 알려주기 위해 사용
  - C, **C++**, Java, Python, ...

5

## C++의 탄생



- C++
  - *Bjarne Stroustrup*
  - C언어의 장점을 유지
  - C with Classes
  - 객체 지향의 개념
  - 최신 프로그래밍 기법 지원
- 표준화
  - ANSI, ISO
  - **C++11**, C++14, C++17
    - ~~MSVC 2010~~
    - MSVC 2013, MSVC 2015

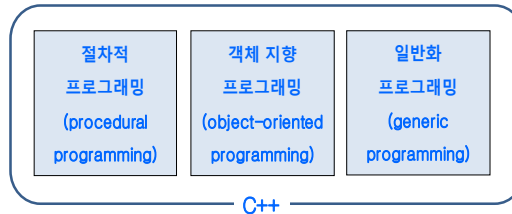


6

## C++의 특징



- C++에서 지원하는 프로그래밍 기법들



- 주요 특징

- C언어 확장: 참조자, 함수중복, 예외처리 등
- 객체지향 지원: 캡슐화(encapsulation), 정보 은닉
- 코드 재사용: 상속 (inheritance), 다형성 (polymorphism)
- 일반화 프로그래밍: 템플릿(template)

7

## 1.2 객체지향 프로그래밍



- 객체지향 프로그래밍과 C++
- Lab: 절차적/구조적/객체지향 비교

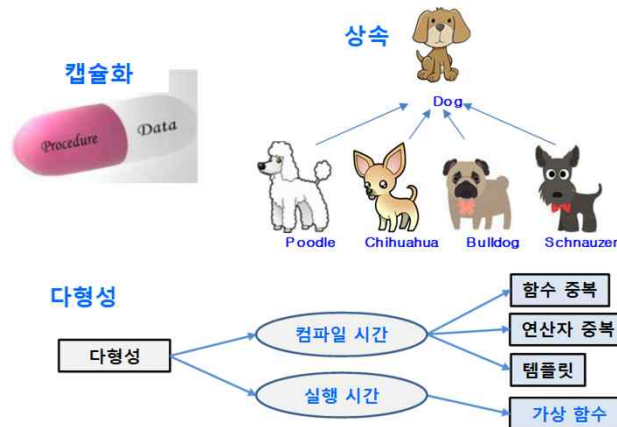
8

## 객체지향 프로그래밍



### • 객체지향의 주요 개념

- 캡슐화
- 정보 은닉
- 상속
- 다형성



9

## Lab: 절차적/구조적/객체지향 비교



- **절차적 프로그래밍(procedural programming)**
  - 데이터보다는 절차, 즉 알고리즘을 중시하는 방법: 어셈블리어
  - goto문이나 jump문
- **구조적 프로그래밍(structured programming)**
  - 절차적 프로그래밍의 하위 개념
  - goto문을 없애거나 goto문에 대한 의존성을 줄임: C언어
- **객체 지향 프로그래밍(Object-Oriented Programming)**
  - 데이터와 알고리즘이 묶임
  - 프로그램: 여러 개의 독립된 "객체"들의 모임으로 인식
  - 소프트웨어 개발과 보수를 간편
  - C++나 Java

10

## (1) 구구단 출력(절차적)



```
#include <stdio.h>
void main()
{
    // 절차적인 프로그래밍
    int i = 1;
    int dan = 3;
    printf("[절차적 프로그래밍]\n");
    printf("[구구단 %d 단]\n", dan);
loop:
    printf("%2d x %2d = %2d\n", dan, i, dan*i);
    i++;
    if( i<=9 ) goto loop;
}
```

```
C.. - □ ×
[절차적 프로그래밍]
[구구단 3 단]
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
```

11

## (2) 구구단 출력(구조적)



```
#include <stdio.h>
void printGuguDan( int dan )
{
    for (int i=1 ; i<=9 ; i++ )
        printf("%2d x %2d = %2d\n", dan, i, dan*i);
}

void main()
{
    // 구조화 프로그래밍
    printf("[구조화 프로그래밍]\n");
    printf("[구구단 %d 단]\n", dan);
    printGuguDan( 3 );
}
```

```
C.. - □ ×
[구조화 프로그래밍]
[구구단 3 단]
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
```

12

### (3) 구구단 출력(객체지향)



```
#include <stdio.h>
class GuguGame {
private:
    int    from;
    int    to;
public:
    GuguGame(){ set(1,9); }
    ~GuguGame(){}
    void set ( int f, int t ) { from = f; to = t; }
    void play ( int dan ) {
        printf("[객체지향 프로그래밍]\n");
        printf("[구구단 %d 단]\n", dan);
        for (int i=from ; i<=to ; i++ ) {
            printf("%2d x %2d = %2d\n", dan, i, dan*i);
        }
    }
};

void main()
{
    GuguGame myGame;
    myGame.play( 3 );
    myGame.set(2,9);
    myGame.play( 3 );
}
```

```
C:\W...  -  □  ×
[객체지향 프로그래밍]
[구구단 3 단]
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
[객체지향 프로그래밍]
[구구단 2 단]
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
```

13

## 1.3 프로그램 개발 과정



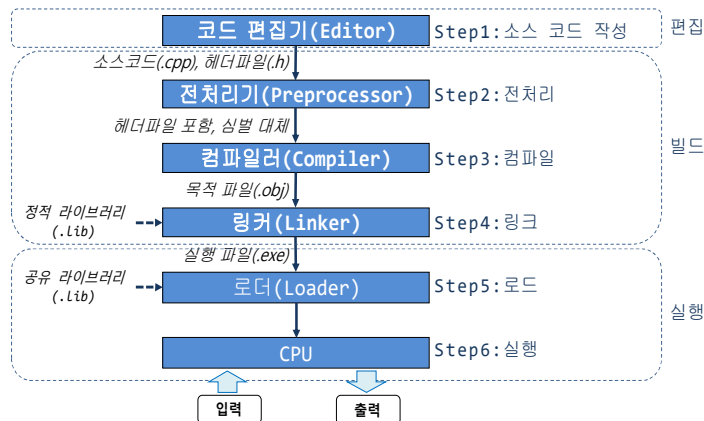
- C++ 프로그램 개발 단계
- 통합 개발 환경

14

# C++ 프로그램 개발 단계



## • C++ 프로그램 개발 단계



15

# 통합 개발 환경

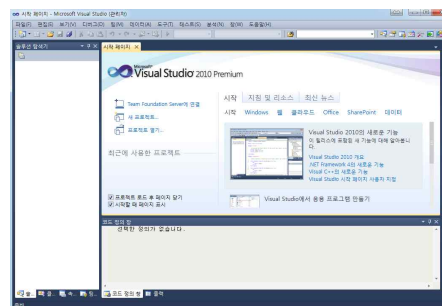


## • 현재

- 통합 개발 환경(Integrated Development Environment, IDE)

## • 과거: 예) 1990년

- 코드 편집 프로그램
  - vi 에디터
- 컴파일과 링크 프로그램
  - gcc(GNU 컴파일러)
- 프로그램 실행
  - 콘솔 환경에서 실행파일 구동
- 디버깅 프로그램
  - gdb(GNU 디버거)



16



## 1.4 프로그램 개발 전의 준비사항



- 준비사항
  - 파일, 폴더, 드라이버, 확장자의 개념을 잘 이해하자.
  - 프로그래밍을 위한 폴더를 준비하자.
  - 프로젝트와 솔루션의 개념을 정확히 이해하자.
  - 모든 이름은 신중하게 정하는 습관을 들이자.
- C++에서는 머리를 쓰는 것이 중요한 것이 아니다. **절차에 익숙해지는 훨씬 중요하다.** 기본적인 절차를 무시하지 않는 습관이 중요하다.

17

## 1.5 C++ 프로그램 개발과정 따라 하기



- Lab

18

## 1.6 소스 코드 분석



```
// GameOverMain.cpp
#include <stdio.h> /* 헤더파일 포함 */
void main()      /* int나 void 반환 */
{
    printf("Game Over !\n"); // '\n' 특수한 의미의 문자
}
```



- **요소들**

- 주석문
- 전처리기 지시자

```
#include <stdio.h> /* C언어에서와 동일한 방법의 헤더파일 포함 */
#include <cstdio> /* C++형태로 변환된 헤더포함(.h를 넣지 않음) */
```

- main() 함수
- 입출력 함수
- 이스케이프 시퀀스

19

## 1.7. 프로그램의 입력과 출력



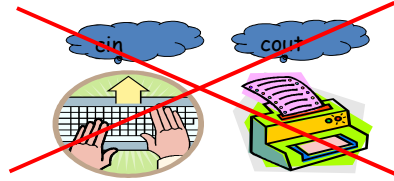
- C++에서의 입출력 방법들
- printf() : 화면 출력 함수
- 이스케이프 시퀀스
- scanf() : 키보드 입력 함수
- Lab: 입출력 예제

20

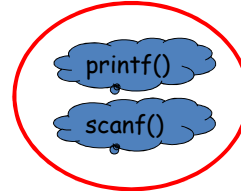
## C++에서의 입출력 방법들



- C++에서의 입출력
- **방법 1:** 키보드 입력은 cin, 콘솔 출력은 cout이 담당
  - 이들은 입출력 객체 (iostream 라이브러리에 포함)
  - 6장 이후



- **방법 2:** C언어의 입출력 함수 사용
  - printf(), scanf() 등의 함수 사용
  - 5장까지는 이 방법을 사용함



21

## printf() : 화면 출력 함수



```
int printf( const char*, ...);
```

- 매개변수의 개수가 달라질 수 있음(매우 어려운 함수)
- 첫 번째 매개변수는 반드시 " "로 이루어진 제어 문자열
  - 출력할 문자열이나 변환 명세, 이스케이프 시퀀스를 표현
  - 변환 명세는 '%'로 시작
  - 변환 명세의 수만큼의 인수가 이후에 반드시 추가되어야 함

```
printf ("C++는 재미있어요!\n");  
printf ("나이는%d\n", age);  
printf ("나이는%d, 키=%fcm\n", age, h);  
printf ("나이 : %3d\n", age);  
printf ("키: %5.1f\n", h);
```

- 제어 문자열에는 이스케이프 시퀀스(escape sequence), 또는 확장 특수문자로 불리는 특수한 문자들이 들어갈 수 있다.

22

## 이스케이프 시퀀스



이스케이프 시퀀스! 설명	표현	
\'	작은 따옴표	byte 0x27
\"	큰 따옴표	byte 0x22
\?	물음표	byte 0x3f
\\	역슬래시	byte 0x5c
\0	공백문자	byte 0x00
\a	부저음	byte 0x07
\b	백스페이스	byte 0x08
\f	폼피드 - 새페이지	byte 0x0c
\n	라인피드 - 줄바꿈	byte 0x0a
\r	캐리지 리턴	byte 0x0d
\t	수평 탭	byte 0x09
\v	수직 탭	byte 0x0b
\nnn	임의의 8진수	byte nnn
\xnn	임의의 16진수	byte nn
\unnnn	임의의 <u>유니코드</u> 값. 몇 개의 문자가 될 수 있음.	code point U+nnnn
\Unnnnnnnn	임의의 <u>유니코드</u> 값. 몇 개의 문자가 될 수 있음.	code point U+nnnnnnnn

23

## scanf() : 키보드 입력 함수



```
int scanf( const char*, ...);
```

- 성공적으로 읽은 매개변수의 수를 반환 (**역시 어려움**)
- 제어 문자열에는 "변환 명세만" 들어감
  - 출력용 문자열이나 이스케이프 시퀀스를 넣지 않아야 함.
  - 변환 명세의 수만큼의 인수가 이후에 반드시 추가되어야 함
  - 두 번째 매개변수부터 값이 아니라 변수의 주소가 들어가야 함

```
int age; // 자료형이 int인 변수 age를 선언
float height; // 자료형이 float인 변수 height를 선언
double weight; // 자료형이 double인 변수 weight를 선언
scanf("%d", &age); // 정수를 읽어 변수 age에 저장
scanf("%f", &height); // float를 읽어 변수 height에 저장
scanf("%lf", &weight); // double을 읽어 변수 weight에 저장
```

24

## Lab: 입출력 예제



```
#include <stdio.h>
void main()
{
    int i;
    double d;
    char str[100];
    printf("정수 입력: ");
    scanf("%d", &i);
    printf(" - 입력된 정수: %d\n", i);
    printf(" - 정수 변수의 주소: %0x\n\n", &i);

    printf("실수 입력: ");
    scanf("%lf", &d);
    printf(" - 입력된 실수: %lf\n", d);
    printf(" - 실수 변수의 주소: %0x\n\n", &d);

    printf("문자열 입력: ");
    scanf("%s", str);
    printf(" - 입력된 문자열: %s\n", str);
    printf(" - 문자열의 주소: %0x\n", str);
}
```

C:\WINDOWS\system32\cmd.exe  
정수 입력: 10  
- 입력된 정수: 10  
- 정수 변수의 주소: effc58

C:\WINDOWS\system32\cmd.exe  
실수 입력: 3.14159  
- 입력된 실수: 3.141590  
- 실수 변수의 주소: effc48

C:\WINDOWS\system32\cmd.exe  
문자열 입력: C++화이팅!!!  
- 입력된 문자열: C++화이팅!!!  
- 문자열의 주소: effbdc  
계속하려면 아무 키나 누르십시오

25

## 1.8 응용: Game Over 게임



- ASCII Art
- Game Over Game Version 2
- Game Over Game Version 3
- 고려해야 할 사항들

26

## ASCII Art



### • 아스키 아트(ASCII Art)

- 아스키 코드(ASCII Code)에 포함된 문자나 기호를 이용한 그림
- Google 검색



27

## Game Over Game Version 2



```
#include <stdio.h>
void main()
{
    printf("
    / / / / / / / / / / / / / / / / \n");
    printf("
    | | | | | | | | | | | | | | | | \n");
    printf("
    | | | | | | | | | | | | | | | | \n");
    printf("
    | | | | | | | | | | | | | | | | \n");
    printf("
    | | | | | | | | | | | | | | | | \n");
    printf("
    | | | | | | | | | | | | | | | | \n");
    printf("
    | | | | | | | | | | | | | | | | \n");
}
```

출력이 좀 이상??? 왜?



```
1>ClCompile:
1> GameOverGame.cpp
1>d: \C++프로그래밍\게임-프로그래밍\기초\gameovergame\gameovergame.cpp(6): warning C4129: ' ' : 문자 이스케이프 시퀀스를 인식할 수 없습니다.
1>d: \C++프로그래밍\게임-프로그래밍\기초\gameovergame\gameovergame.cpp(6): warning C4129: ' ' : 문자 이스케이프 시퀀스를 인식할 수 없습니다.
1>d: \C++프로그래밍\게임-프로그래밍\기초\gameovergame\gameovergame.cpp(8): warning C4129: ' ' : 문자 이스케이프 시퀀스를 인식할 수 없습니다.
1>d: \C++프로그래밍\게임-프로그래밍\기초\gameovergame\gameovergame.cpp(8): warning C4129: ' ' : 문자 이스케이프 시퀀스를 인식할 수 없습니다.
```

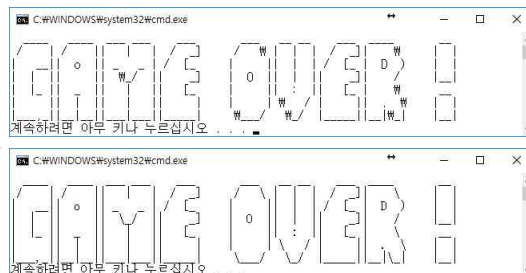
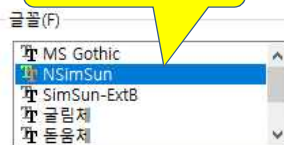
28

# Game Over Game Version 3

[illegible]

이스케이프 시퀀스 처리 !!!  
₩ 문자는???

## 폰트 변경 !!!



29

- 핵심 고려 사항

- 이스케이프 시퀀스 고려
- 출력할 문자열의 길이 (콘솔 창 한 줄의 길이)
- 콘솔창의 글꼴: **가변폭 폰트/고정폭 폰트**
- 편집기 사용 팁: **영역 복사** **ALT**
- 솔루션 - 프로젝트



30



- 몇 가지 추가적인 의문

- 그림을 출력하려면?
- printf()문장에 직접 넣지 않는 방법?
  - #define
  - 파일 입출력
- 자동화 할 수 있는 방법은 없을까?
- 한글을 처리할 수 있는 방법은?

31

## 1장 요약문제, 연습문제, 실습문제



32





감사합니다!