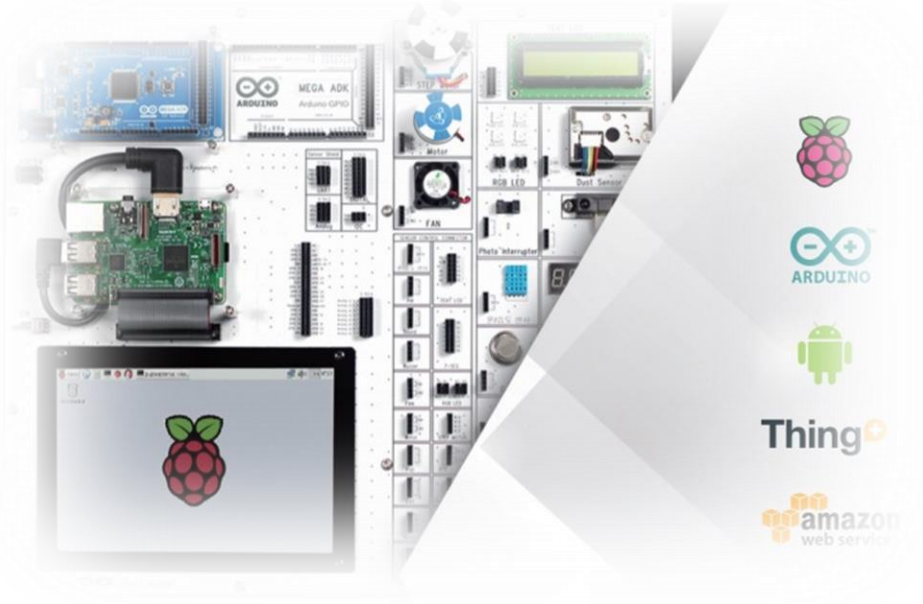


2021년도 2학기

휴먼스마트기기설계



교과목명 : 휴먼스마트기기설계 (01)

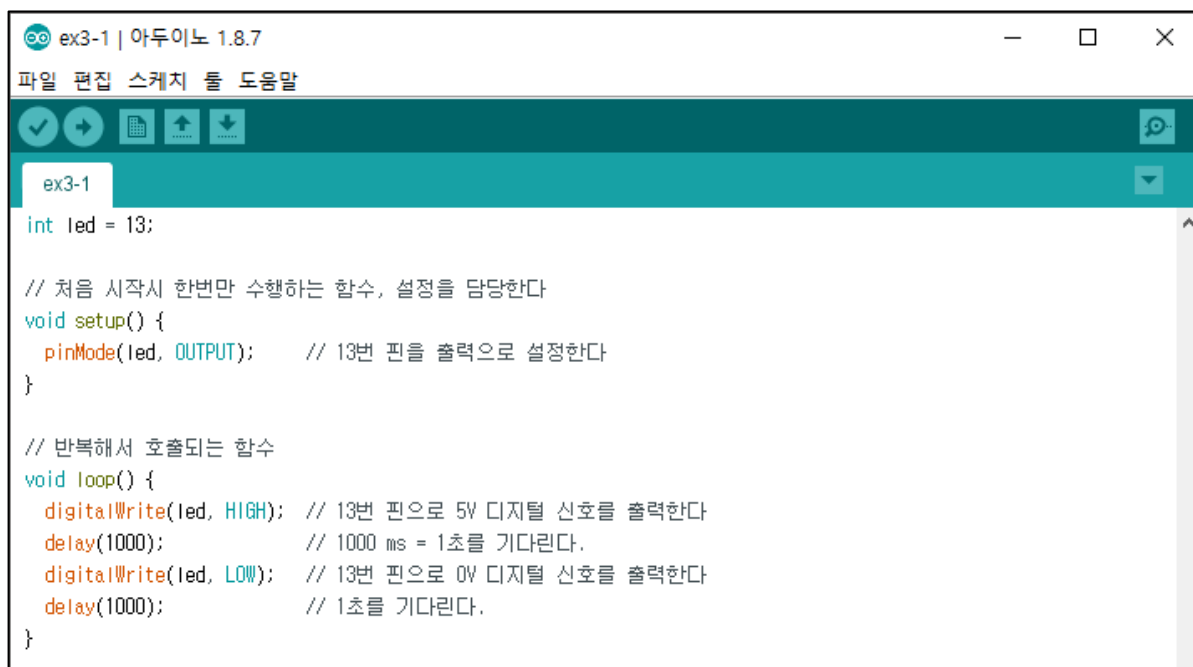
담당교수 : 이 수 형

E-mail : soohyong@uu.ac.kr

교재명 : 스마트기기 개발을 위한 사물인터넷, 이수형. (LINC+ 사업단 배포)

A. C++기초

※ 참고 : 2019학년도 시스템응용프로그래밍 I,
교재 : 명품 C++ Programming, 황기태, 생능출판사

A screenshot of the Arduino IDE interface. The title bar reads "ex3-1 | 아두이노 1.8.7". The menu bar includes "파일", "편집", "스케치", "툴", and "도움말". The toolbar contains icons for checking, running, saving, and uploading. The file explorer on the left shows a tab for "ex3-1". The main editor area contains the following C++ code:

```
int led = 13;

// 처음 시작시 한번만 수행하는 함수, 설정을 담당한다
void setup() {
  pinMode(led, OUTPUT);    // 13번 핀을 출력으로 설정한다
}

// 반복해서 호출되는 함수
void loop() {
  digitalWrite(led, HIGH); // 13번 핀으로 5V 디지털 신호를 출력한다
  delay(1000);             // 1000 ms = 1초를 기다린다.
  digitalWrite(led, LOW);  // 13번 핀으로 0V 디지털 신호를 출력한다
  delay(1000);             // 1초를 기다린다.
}
```

A.1 C++ 언어란?

- C++ 언어

- C언어의 기능

- 대부분의 기능이 호환됨, 좀 더 엄격하게 바뀐 부분이 일부 있음

- 객체지향 프로그래밍 (Object Oriented Programming) ⇒ 클래스

- 캡슐화(encapsulation), 상속성(inheritance), 다형성(polymorphism)

- 더욱 편리하게 확장된 문법

- 인라인 함수 : 함수 호출 대신 함수 코드의 삽입

- 함수 중복 (function overloading) : 동일한 이름의 인수의 개수/타입이 다른 여러 개의 함수

- 디폴트 매개변수(default parameter) : 미리 지정한 인수값 생략 가능

- 참조 (reference) : 기존의 변수의 다른 이름 지정

- 참조에 의한 호출 : 인수로 참조 전달 → 포인터 변수의 단점 해결 가능

- new/delete 연산자 : 동적 메모리 할당/해제

- 연산자 재정의 (operator overriding) : 임의의 객체 타입(클래스)에 연산자 사용 가능

A.1 C++ 언어란?

- 기타 C vs. C++
 - printf() 함수 대신 cout 객체 사용
 - `printf("Hello\n");` ⇒ `cout << "Hello\n";`
 - 한줄 주석문 추가 (//)
 - // 이후의 문장이 모두 주석 처리됨
 - 함수를 사용하기 전에 반드시 함수의 선언이 있어야 함
 - C에서는 경고, C++에서는 오류 ⇒ 더 엄격해짐
 - 실행문 중간에 변수 정의 가능
 - C언어에서는 블록(보통 함수)의 제일 위부분에만 변수 정의/선언 가능
 - C++에서는 사용하는 위치에서 변수 정의 ⇒ 가독성 향상 및 타이핑 오류 감소
 - STL 과 같은 다양한 라이브러리 및 템플릿(template) 사용 가능

A.2 C++ 확장 문법

- 함수 중복 (다형성)

- 하나의 이름, 여러 개의 함수, 인수의 개수/타입으로 구분

```
int sum(int a, int b, int c) {  
    return a + b + c;  
}  
  
double sum(double a, double b) {  
    return a + b;  
}  
  
int sum(int a, int b) {  
    return a + b;  
}
```

성공적으로 중복된 sum() 함수들

```
int main(){  
    cout << sum(2, 5, 33);  
  
    cout << sum(12.5, 33.6);  
  
    cout << sum(2, 6);  
}
```

중복된 sum() 함수 호출.
컴파일러가 구분

A.2 C++ 확장 문법

- 디폴트 매개 변수

- 인수의 값을 지정하지 않은 경우, 미리 지정된 인수값으로 대체함

```
void star(int a=5); // a의 디폴트 값은 5
```

```
star(); // 매개 변수 a에 디폴트 값 5가 전달됨. star(5);와 동일  
star(10); // 매개 변수 a에 10을 넘겨줌
```

- 디폴트 매개 변수는 뒤쪽의 인수에만 사용가능함 → 모호성 방지

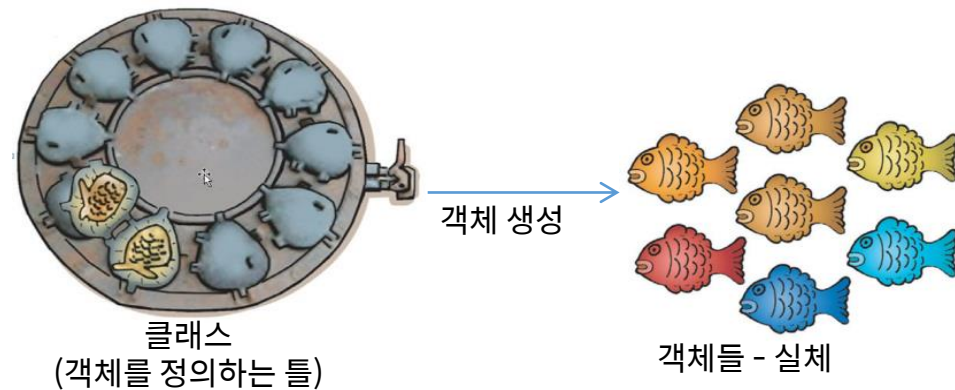
컴파일 오류

```
void calc(int a, int b=5, int c, int d=0); // 컴파일 오류  
void sum(int a=0, int b, int c); // 컴파일 오류
```

```
void calc(int a, int b=5, int c=0, int d=0); // 컴파일 성공
```

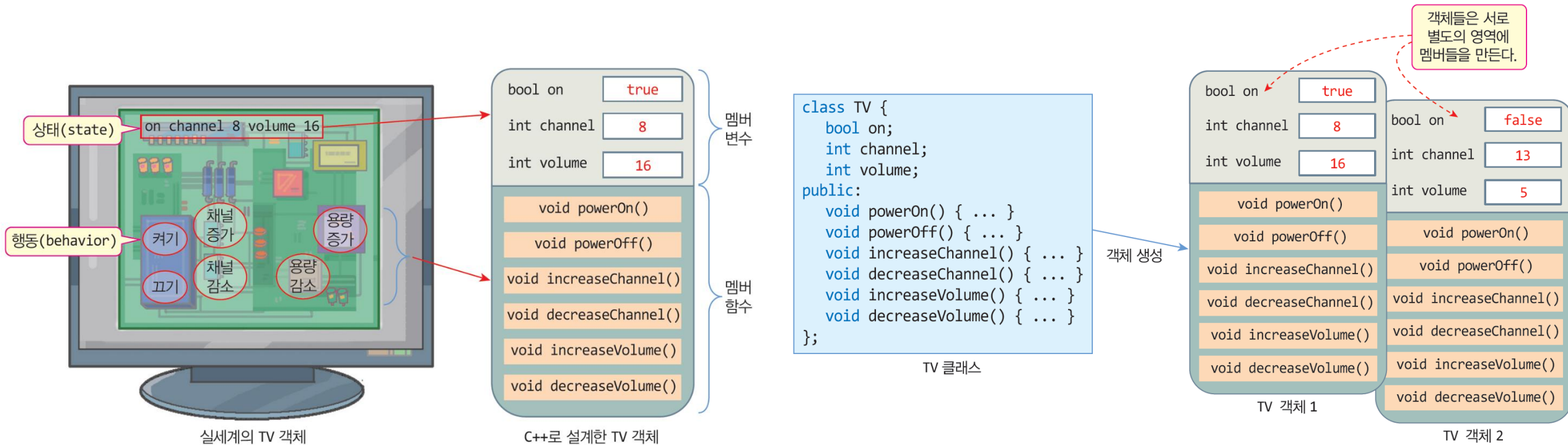
A.3 클래스

- 클래스 (class)
 - C 언어의 구조체(structure)의 확장
 - 객체를 만들어내기 위한 설계도, 틀 (실제 메모리를 점유하지 않음)
 - 멤버 변수와 멤버 함수로 구성
- 객체 (object, instance, ...)
 - 클래스를 이용하여 실제 구현(만들어진) 것 : 변수와 비슷한 개념
 - 내부에 클래스에서 선언한 멤버 변수/함수들이 포함됨
 - 각 객체는 독립된 메모리를 점유함



A.3 클래스 : OOP

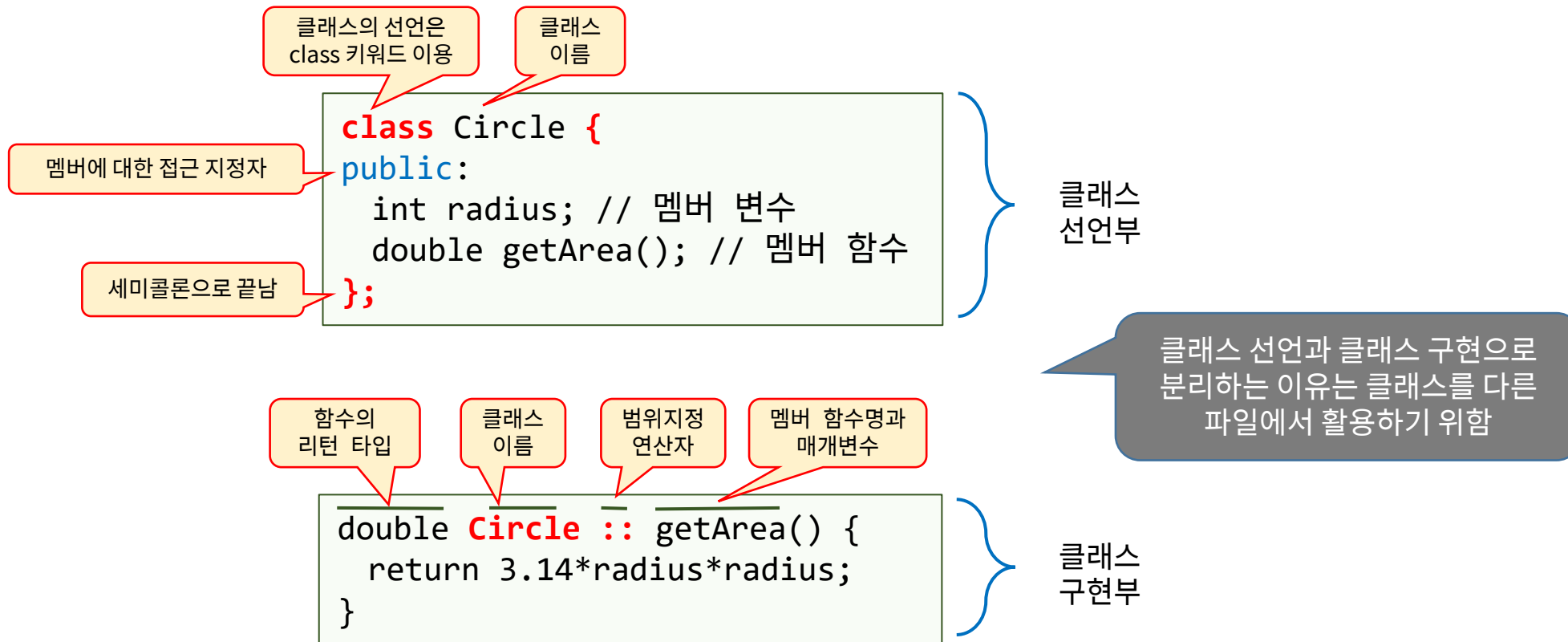
- TV의 추상화 : C++ 에서 TV를 다루기 위한 프로그래밍 방법



(b) C++로 표현한 TV 클래스와 TV 객체들

A.2 클래스 정의

• 클래스의 정의



A.2 클래스 객체 정의 및 사용

• 객체의 생성 및 사용

- 객체의 이름 및 객체 생성

```
Circle donut; // 이름이 donut 인 Circle 타입의 객체 생성
```

객체의 타입.
클래스 이름

객체 이름

객체가 생성되면
메모리가 할당된다.

```
int radius   
double getArea() {...}
```

donut 객체

- 객체의 멤버 변수 접근 : 멤버 연산자 (.) 사용

```
donut.radius = 1; // donut 객체의 radius 멤버 값을 1로 설정
```

객체 이름

멤버 변수

객체 이름과
멤버 사이에
. 연산자

```
int radius   
double getArea() {...}
```

donut 객체

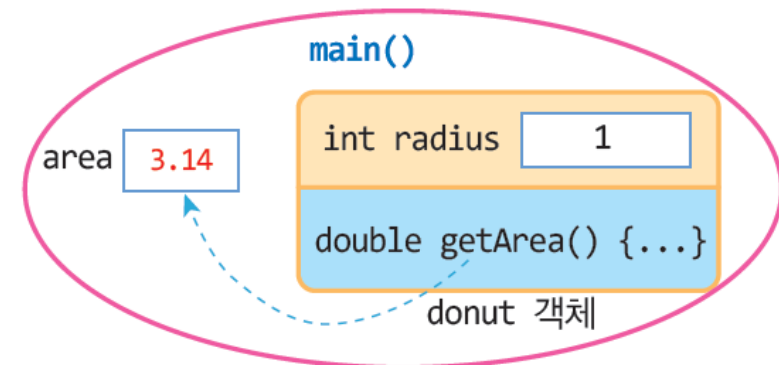
- 객체의 멤버 함수의 접근 : 멤버 연산자 사용

```
double area = donut.getArea(); // donut 객체의 면적 알아내기
```

객체 이름

멤버 함수 호출

객체 이름과 멤버 사이에
. 연산자



A.2 클래스 객체 정의 및 사용

```
#include <iostream>
using namespace std;

class Circle {
public:
    int radius;
    double getArea();
};

double Circle::getArea() {
    return 3.14*radius*radius;
}

int main() {
    Circle donut;
    donut.radius = 1; // donut 객체의 반지름을 1로 설정
    double area = donut.getArea(); // donut 객체의 면적 알아내기
    cout << "donut 면적은 " << area << endl;

    Circle pizza;
    pizza.radius = 30; // pizza 객체의 반지름을 30으로 설정
    area = pizza.getArea(); // pizza 객체의 면적 알아내기
    cout << "pizza 면적은 " << area << endl;
}
```

객체 donut
생성

donut의 멤버
변수 접근

donut의 멤버
함수 호출

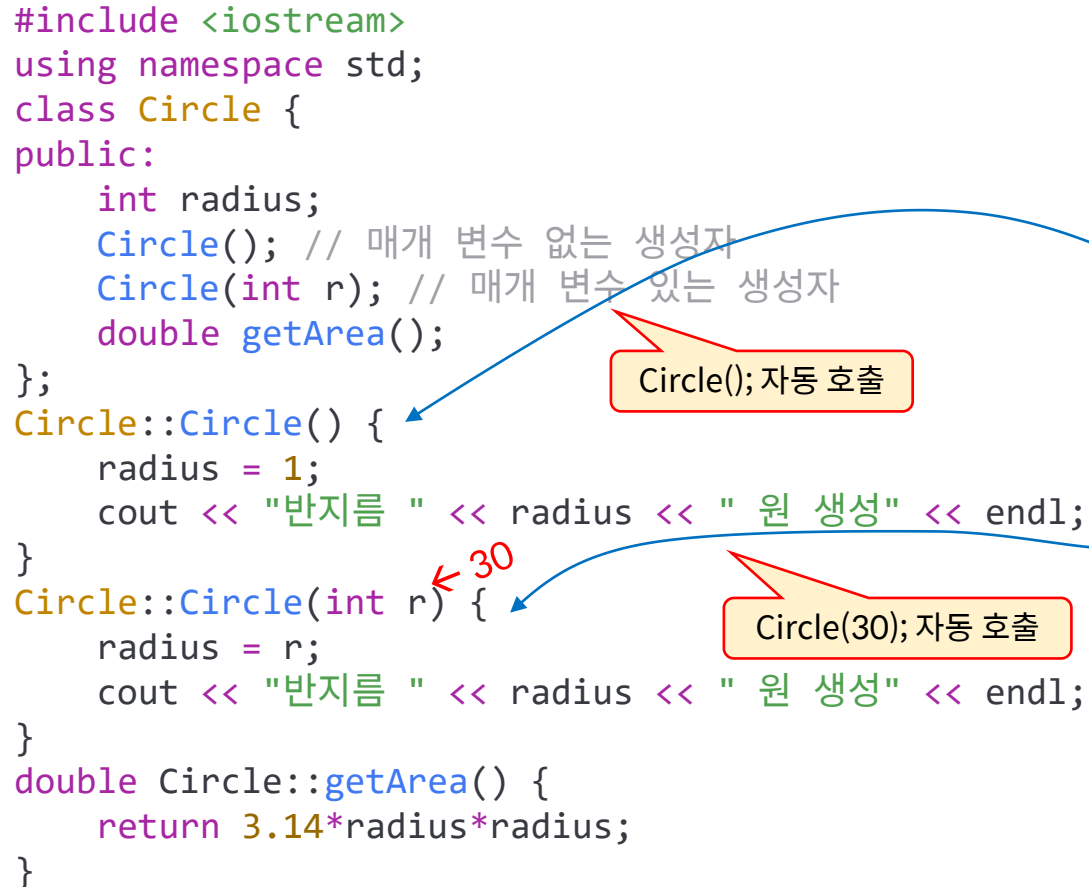
donut 면적은 3.14
pizza 면적은 2826

A.2 클래스 : 생성자

- 생성자(constructor)

- 객체가 생성되는 시점에 자동으로 호출되는 함수 : 주로 객체의 초기화 담당

```
#include <iostream>
using namespace std;
class Circle {
public:
    int radius;
    Circle(); // 매개 변수 없는 생성자
    Circle(int r); // 매개 변수 있는 생성자
    double getArea();
};
Circle::Circle() {
    radius = 1;
    cout << "반지름 " << radius << " 원 생성" << endl;
}
Circle::Circle(int r) {
    radius = r;
    cout << "반지름 " << radius << " 원 생성" << endl;
}
double Circle::getArea() {
    return 3.14*radius*radius;
}
```



```
int main() {
    Circle donut; // 매개 변수 없는 생성자 호출
    double area = donut.getArea();
    cout << "donut 면적은 " << area << endl;

    Circle pizza(30); // 매개 변수 있는 생성자 호출
    area = pizza.getArea();
    cout << "pizza 면적은 " << area << endl;
}
```

반지름 1 원 생성
donut 면적은 3.14
반지름 30 원 생성
pizza 면적은 2826

A.2 클래스 : 상속 (Inheritance)

- 상속 : 하나의 클래스를 상속받아 기능을 추가한 새로운 클래스 생성
 - 기본 클래스 (base class) + α $\Rightarrow$ 파생 클래스 (derived class)
부모 클래스 (parent class) 자식 클래스 (child class)



C++로 상속 선언

```
class Phone {
    void call();
    void receive();
};

class MobilePhone : public Phone {
    void connectWireless();
    void recharge();
};

class MusicPhone : public MobilePhone {
    void downloadMusic();
    void play();
};
```

Phone을 상속받는다.

MobilePhone을 상속받는다.



전화기



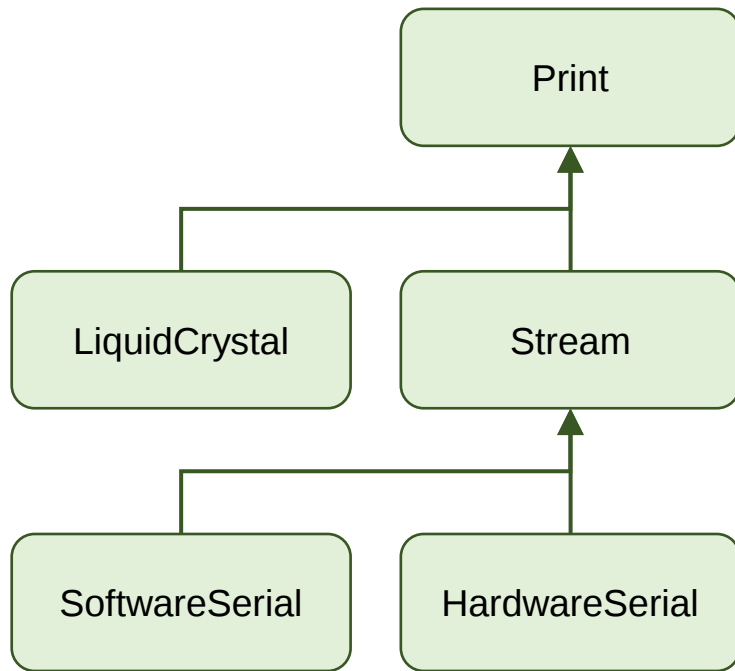
휴대 전화기



음악 기능
전화기

A.2 클래스 : 상속

- 아두이노에서 상속 클래스 사용 예
 - Serial : HardwareSerial 클래스의 객체
 - HardwareSerial 클래스 : Stream 클래스를 상속함



HardwareSerial.h 의 일부 (아두이노 프로그램의 내부 헤더파일)

```
class HardwareSerial : public Stream
{
protected:
public:
    inline HardwareSerial(
        volatile uint8_t *ubrrh, volatile uint8_t *ubrrl,
        volatile uint8_t *ucsrA, volatile uint8_t *ucsrB,
        volatile uint8_t *ucsrc, volatile uint8_t *udr);
    void begin(unsigned long baud)
        { begin(baud, SERIAL_8N1); }
    void begin(unsigned long, uint8_t);
    void end();
    virtual int available(void);
};

// ...
extern HardwareSerial Serial;
```