



함수와 참조, 복사 생성자

학습 목표

1. 값에 의한 호출과 주소에 의해 호출을 복습한다.
2. 함수 호출 시 객체가 전달되는 과정을 이해한다.
3. 객체 치환과 객체 리턴을 이해한다.
4. 참조에 대한 개념을 이해하고, 참조 변수를 선언할 수 있다.
5. 참조에 의한 호출과 참조 리턴에 대해 이해하고 코드를 작성할 수 있다.

참조에 의한 호출로 객체 전달

3

- 참조에 의한 호출로 객체 전달
 - ▣ 함수의 매개변수를 참조 객체로 선언
 - ▣ 매개변수 참조이므로 이름만 생성되고 공간할당 안됨 -> 매개변수객체의 생성자와 소멸자는 실행되지 않음
 - ▣ 인수객체로 초기화 되면 매개변수 객체는 인수(원본)객체의 별명이 됨
 - ▣ 참조 매개변수(객체)에 대한 모든 연산은 인수로 전달된 원본 객체에 대한 연산이 된다.
 - ▣ 다른 함수에서 선언된 객체의 값을 변경하고자 할 때 사용(주소에 의한 호출을 사용해야 하는 경우)

예제 5-6 참조에 의한 호출로 Circle객체에 참조 전달

4

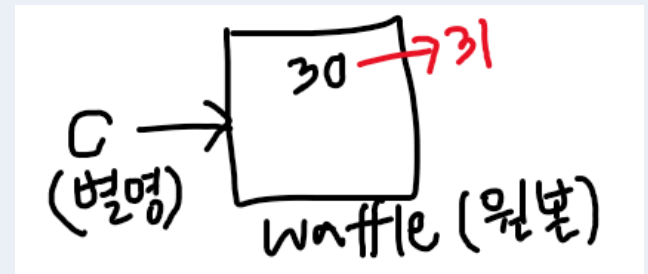
```
#include <iostream>
using namespace std;
class Circle {
private:
    int radius;
public:
    Circle();
    Circle(int r);
    ~Circle();
    double getArea() { return 3.14 * radius * radius; }
    int getRadius() { return radius; }
    void setRadius(int radius) { this->radius = radius; }
};
Circle::Circle() {
    radius = 1;
    cout << "생성자 실행 radius = " << radius << endl;
}
```

예제5-6 참조에 의한 호출로 Circle객체에 참조 전달

5

```
Circle::Circle(int radius) {  
    this->radius = radius;  
    cout << "생성자 실행 radius = " << radius  
        << endl;  
}  
Circle::~~Circle() {  
    cout << "소멸자 실행 radius=" << radius << endl;  
}  
void increaseCircle(Circle& c) { // 매개변수객체 생성자와 소멸자 호출안됨  
    int r = c.getRadius();  
    c.setRadius(r + 1);  
}  
int main() {  
    Circle waffle(30);  
    increaseCircle(waffle);  
    cout << waffle.getRadius() << endl;  
}
```

생성자 실행 radius = 30
31
소멸자 실행 radius = 31



예제5-7 참조 매개 변수를 가진 함수 만들기 연습

6

- 키보드로부터 반지름 값을 읽어 Circle 객체에 반지름을 설정하는 readRadius() 함수를 작성하라.

```
#include <iostream>
using namespace std;
class Circle {
    int radius;
public:
    Circle() { radius = 1; }
    Circle(int radius) { this->radius = radius; }
    void setRadius(int radius) { this->radius = radius; }
    double getArea() { return 3.14 * radius * radius; }
};
```

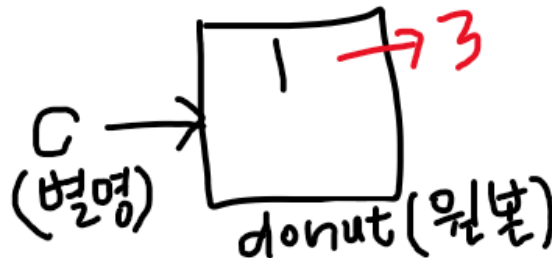
정수 값으로 반지름을 입력하세요>>3<엔터>
donut의 면적 = 28.26

예제 5-7 정답

7

```
void readRadius(Circle& c) {  
    int r;  
    cout << "정수 값으로 반지름을 입력하세요>>";  
    cin >> r;           // 반지름 값 입력  
    c.setRadius(r);     // 객체 c에 반지름 설정  
}  
  
int main() {  
    Circle donut;  
    readRadius(donut);  
    cout << "donut의 면적 = " << donut.getArea() << endl;  
}
```

정수 값으로 반지름을 입력하세요>>3<엔터>
donut의 면적 = 28.26



참조 리턴

8

□ C 언어의 함수 리턴

▣ 함수는 반드시 값만 리턴

- 기본 타입 값 : int, char, double, 구조체, 객체 등에 저장된 값
- 주소값(반환형이 포인터로 선언된 경우)
- 리턴값은 상수로 취급됨

□ C++의 함수 리턴

▣ 함수는 값 외에 참조(별명) 리턴 가능 -> 함수의 반환형이 참조

▣ 참조(별명) 리턴

- 변수, 객체와 같이 존재하는 변수에 대한 참조(별명)을 리턴
- 변수의 값을 리턴하는 것이 아니고 변수의 별명을 리턴
- 리턴값은 변수로 취급됨
- 참조 = 별명 = 변수 = 공간

함수의 return문 실행과정

9

- 함수가 리턴할 때 임시변수가 할당되고 리턴값으로 초기화 -> 함수 호출문장은 임시변수(반환값)로 치환됨

```
int main(void)
{
    int num = 10, result;
    result = simplefn(num);
}

int simplefn(int n)
{
    int m;
    ...
    return m;
}
```

// result = 임시변수;
// 함수호출문이 임시변수로 대체

임시변수
(이름없음) // 컴파일러가 자동으로 추가하는 임시변수

// 임시변수 할당과 반환값으로 초기화
→ int 임시변수 = m;

함수의 return문 실행과정

10

- 함수호출문은 임시변수로 대체됨 -> 대입연산 실행 -> 임시변수 사라짐 -> 다음 문장 실행

```
result = simplefn(num);
```



```
result = 임시변수;
```



임시변수는 메모리에서 사라짐



다음 문장 실행됨 -> 임시변수사용불가

값을 리턴하는 함수 vs. 참조를 리턴하는 함수

11

```
char c = 'a';  
char get() {  
    return c;  
}  
  
char a = get();  
get() = 'b';  
  
// char값을 리턴  
// 문자 상수값('a')을 리턴 -> 상수취급  
  
// a = 'a'가 됨, 정상  
// 컴파일 에러  
// 'a' = 'b' 와 같음, 좌변에 상수가 오면 안됨
```

(a) 문자 값을 리턴하는 get()

값을 리턴하는 함수 vs. 참조를 리턴하는 함수

12

```
char c = 'a';  
char& find() {  
    return c;  
}  
  
char a = find();  
char& ref = find();  
ref = 'M';  
find() = 'b';
```

// char 타입의 참조 리턴
// 변수 c의 참조(별명)를 리턴, 변수취급

// a = 'a'가 됨
// ref는 c에 대한 참조(별명)
// c = 'M'
// c = 'b' 와 같음
// 좌변->변수c에 대한 참조(공간)->c(변수)

(b) char 타입의 참조(공간)을 리턴하는 find()

예제 5-8 간단한 참조 리턴 사례

13

```
#include <iostream>
using namespace std;
char& find(char s[], int index) {
    return s[index];
}

int main() {
    char name[] = "Mike";
    cout << name << endl;
    find(name, 0) = 'S';
    cout << name << endl;
    char& ref = find(name, 2);
    ref = 't';
    cout << name << endl;
}
```

// s[index] 공간의 참조 리턴

// name[0]='S'로 변경

// ref는 name[2] 참조, name[2] = 't'

Mike
Sike
Site

예제 5-8 간단한 참조 리턴 사례

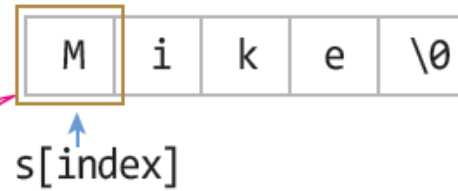
14

(1) `char name[] = "Mike";`



(2) `return s[index];`

공간에 대한
참조, 즉 익명
의 이름 리턴



(3) `find(name, 0) = 'S';`



(4) `ref = 't';`



실습과제1

15

- C언어 함수의 리턴과정을 상세히 설명하라.
- 반환형이 참조로 선언된 경우 리턴값의 의미를 설명하라.

실습과제2

16

- 매개변수로 받은 string객체의 내용물을 대문자로 변환하는 함수 Uppercase를 참조에 의한 호출을 이용하여 작성하시오. 대문자 변환 시 std::toupper함수를 사용할 것

```
#include <iostream>
#include <string>
using namespace std;
// 함수 선언 추가
int main() {
    string s1("hello");
    cout << "변환전 문자열 " << s1 << endl;
    // 함수 호출
    cout << "변환후 문자열 " << s1 << endl;
    return 0;
}
// 함수 정의 추가
```

변환전 문자열 *hello*
변환후 문자열 *HELLO*

실습과제3

17

- 매개변수를 참조로 선언하여 두 문자열을 교환하는 함수 SwapString 을 작성하시오. 매개변수가 포인터인 경우와 차이점을 설명하라.

```
#include <iostream>
#include <string>
using namespace std;
// 함수 선언 추가
int main() {
    string s1("hello");
    string s2("world");
    cout << "교환전 문자열 " << s1 << s2 << endl;
    // 함수 호출
    cout << "교환후 문자열 " << s1 << s2 << endl;
    return 0;
}
// 함수 정의 추가
```

교환전 문자열 *hello world*
교환후 문자열 *world hello*

과제 제출 방법

18

- 소스코드, 라인단위의 주석, 실행결과를 포함하는 pdf파일을 작성한 후 eclass 과제 게시판에 업로드, **반드시 하나의 pdf파일로 업로드할 것**
- 기한 : 과제 게시판에 마감시간 참조
- 실행결과를 캡처할 때 글자를 알아보기 쉽게 확대해서 캡처할 것.
- 소스코드의 첫 부분은 아래처럼 제목,날짜,작성자(학번,이름)를 작성할 것

```
// *****  
//   제   목   : 정수 4개의 평균을 구하는 프로그램  
//   날   짜   : 2023년 9월10일  
//   작성자   : 15010101 홍길동  
// *****  
  
// 소스코드 작성
```