



객체 포인터와 객체 배열, 객체의 동적 생성

연습 - 이론문제

* 1~5번 문제에 사용되는 Rect 클래스. Rect 클래스는 폭과 높이로 사각형을 추상화한다.

```
class Rect {
    int width, height;
public:
    Rect(int w, int h) { width = w; height = h; }
    int getWidth() { return width; }
    int getHeight() { return height; }
    int getArea();
};

int Rect::getArea() {
    return width*height;
}
```

1. Rect의 객체를 다루는 다음 코드를 작성하려고 한다. 아래의 문제에 따라 빈칸에 적절한 코드를 삽입하라.

```
int main() {
    Rect r(2,3);
    _____(1)
    _____(2)
    _____(3)
}
```

- (1) Rect 클래스에 대한 포인터 변수 p를 선언하라.
- (2) 선언된 포인터 변수 p에 객체 r의 주소를 지정하라.
- (3) 포인터 변수 p를 이용하여 객체 r의 폭과 높이를 출력하라.


```

class Rect {
    int width, height;
public:
    Rect(int w, int h) { width = w; height = h; }
    int getWidth() { return width; }
    int getHeight() { return height; }
    int getArea();
};

int Rect::getArea() {
    return width*height;
}

```

2. 사용자로부터 폭과 높이 값을 입력받아 동적으로 Rect 객체를 생성하고 면적을 구하여 출력하는 코드를 작성하고자 한다. 다음 물음에 따라 빈칸을 채워라.

```

int main() {
    Rect *q;
    int w, h;
    cin >> w >> h; // 사용자로부터 사각형의 폭과 높이를 w, h에 각각 입력받는다.

    _____(1)
    _____(2)
    _____(3)

}

```

- (1) 포인터 변수 q에 wxh 크기의 사각형을 표현하는 Rect 객체를 동적으로 생성한다.
- (2) 포인터 q를 이용하여 사각형의 면적을 출력한다.
- (3) 생성한 객체를 반환한다.

```
class Rect {  
    int width, height;  
public:  
    Rect(int w, int h) { width = w; height = h; }  
    int getWidth() { return width; }  
    int getHeight() { return height; }  
    int getArea();  
};  
  
int Rect::getArea() {  
    return width*height;  
}
```

3. Rect 객체나 배열을 생성하는 다음 코드 중 컴파일 오류가 발생하는 것은?

- ① Rect a;
- ② Rect b(5, 6);
- ③ Rect c[2] = { Rect(1, 1), Rect(2, 3) };
- ④ Rect d[2][3] = { {Rect(1,2), Rect(2,3), Rect(3,4)}, {Rect(1,1), Rect(2,2), Rect(3,3)} };

4. Rect 객체의 배열을 생성하는 다음 코드는 컴파일 오류가 발생한다. 컴파일 오류가 발생하지 않기 위해 Rect 클래스를 어떻게 수정하여야 하는가?

```
Rect *p = new Rect[10];
```

```
class Rect {  
    int width, height;  
public:  
    Rect(int w, int h) { width = w; height = h; }  
    int getWidth() { return width; }  
    int getHeight() { return height; }  
    int getArea();  
};  
  
int Rect::getArea() {  
    return width*height;  
}
```

5. Rect 클래스에 다음과 같은 기본 생성자를 삽입하고,

```
Rect() { width = 1; height = 1; }
```

다음 배열 r 생성 후, 배열 r의 사각형 면적의 합을 출력하는 코드를 작성하라.

```
Rect r[5] = { Rect(), Rect(2, 3), Rect(3,4), Rect(4,5), Rect(5,6) };
```

6. public 속성의 getVolume() 멤버 함수를 가진 Cube 클래스에 대해, 다음 코드가 있다.

```
Cube c;  
Cube *p = &c;
```

다음 중 컴파일 오류가 발생하는 것은?

- ① c.getVolume();
- ② p->getVolume();
- ③ (*p).getVolume();
- ④ c->getVolume();

7. 다음 객체 배열에 관해 잘못 설명된 것은?

```
Cube c[4];
```

- ① 배열 `c`가 생성될 때 `c[0]`, `c[1]`, `c[2]`, `c[3]`의 4개의 `Cube` 객체가 생성된다.
- ② 기본 생성자 `Cube()`가 4번 호출된다.
- ③ 배열 `c`가 소멸될 때 `c[3]`, `c[2]`, `c[1]`, `c[0]`의 순서로 소멸자가 실행된다.
- ④ `delete c`; 코드로 배열 `c`를 소멸한다.

8. 다음 프로그램이 실행될 때 출력되는 결과는 무엇인가?

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

class Color {
    string c;
public:
    Color() { c = "white"; cout << "기본생성자" << endl; }
    Color(string c) { this->c = c; cout << "매개변수생성자" << endl; }
    ~Color() { cout << "소멸자" << endl; }
};

class Palette {
    Color *p;
public:
    Palette() { p = new Color[3]; }
    ~Palette() { delete [] p; }
};

int main() {
    Palette *p = new Palette();
    delete p;
}
```

9. new와 delete는 무엇인가?

① C++의 기본 연산자

② C++ 표준 함수

③ C++의 표준 객체

④ C++의 특수 매크로

10. 다음 코드의 문제점은 무엇인가?

```
Cube *p = new Cube [4];  
delete p;
```

11. `this`에 대해 잘못 말한 것은?

- ① `this`는 포인터이다.
- ② `this`는 컴파일러에 의해 묵시적으로 전달되는 매개 변수이다.
- ③ `this`는 `static` 함수를 포함하여 멤버 함수 내에서만 다루어지는 객체 자신에 대한 포인터이다.
- ④ 연산자 중복에서 `this`가 필요하다.

12. `this`의 활용에 대해 잘못 설명한 것은?

- ① `this`는 클래스의 멤버 함수 외의 다른 함수에서는 사용할 수 없다.
- ② `this`는 `static` 멤버 함수에는 사용할 수 없다.
- ③ `this`는 생성자에서 사용할 수 없다.
- ④ 어떤 멤버 함수에서는 `this`를 리턴하기도 한다.

13. `this`를 최대한 많이 활용하여 다음 클래스를 가장 바람직하게 수정하라.

```
class Location {
    int width, height;
public:
    Location() { width = height = 0; }
    Location(int w, int h) {
        width = w; height = h;
    }
    void show();
};
void Location::show() {
    cout << width << height << endl;
}
```

14. 메모리 누수란 어떤 상황에서 발생하는가?

15. 함수 f()가 실행되고 난 뒤 메모리 누수가 발생하는지 판단하고 메모리 누수가 발생하면 발생하지 않도록 수정하라.

(1)

```
void f() {  
    char *p = new char [10];  
    strcpy(p, "abc");  
}
```

(2)

```
void f() {  
    int *p = new int;  
    int *q = p;  
    delete q;  
}
```

(3)

```
int f() {  
    int n[10] = { 0 };  
    return n[0];  
}
```

(4)

```
void f() {  
    int *p;  
    for(int i=0; i<5; i++) {  
        p = new int;  
        cin >> *p;  
        if(*p % 2 == 1) break;  
    }  
    delete p;  
}
```


16. string 클래스를 사용하기 위해 필요한 헤더 파일은 무엇인가?

- ① <string> ② <string.h> ③ <cstring> ④ <iostream>

17. string s1 = "123"; string s2 = "246"; 일 때, a와 b의 문자열 속에 있는 수를 더 하여 369를 출력하고자 한다. 아래 빈칸을 채워라.

```
int n = _____ (s1);  
int m = _____ (s2);  
cout << n + m;
```

18. 문자열을 다루고자 한다. C-스트링과 string 클래스에 대해 설명이 틀린 것은?

- ① C-스트링은 문자의 배열을 이용하여 문자열을 표현한다.
- ② string 클래스가 문자열을 객체화하므로 C-스트링보다 사용하기 쉽다.
- ③ string 클래스가 좋기는 하지만 C++의 표준이 아니므로 가급적 사용하지 않는 것이 좋다.
- ④ string 클래스는 문자열만 다루지 대문자를 소문자로 변환하는 등 문자를 조작하는 기능은 없다.

19. 다음 프로그램의 각 라인을 string 클래스에서 제공하는 연산자를 이용하여 고쳐라.

```
string a("My name is Jane.");  
char ch = a.at(2);  
if(a.compare("My name is John.") == 0) cout << "same";  
a.append("~~");  
a.replace(1, 1, "Y");
```

연습 - 실습문제

- 1.★ 다음은 색의 3요소인 red, green, blue로 색을 추상화한 Color 클래스를 선언하고 활용하는 코드이다. 빈칸을 채워라. red, green, blue는 0~255의 값만 가진다.

난이도 4

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

class Color {
    int red, green, blue;
public:
    Color() { red = green = blue = 0; }
    Color(int r, int g, int b) { red = r; green = g; blue = b; }
    void setColor(int r, int g, int b) { red = r; green = g; blue = b; }
    void show() { cout << red << ' ' << green << ' ' << blue << endl; }
};

int main() {
    Color screenColor(255, 0, 0); // 빨간색의 screenColor 객체 생성
    Color *p;                      // Color 타입의 포인터 변수 p 선언
    _____ // (1) p가 screenColor의 주소를 가지도록 코드 작성
    _____ // (2) p와 show()를 이용하여 screenColor 색 출력
    _____ // (3) Color의 일차원 배열 colors 선언. 원소는 3개
    _____ // (4) p가 colors 배열을 가리키도록 코드 작성

    // (5) p와 setColor()를 이용하여 colors[0], colors[1], colors[2]가
    // 각각 빨강, 초록, 파랑색을 가지도록 코드 작성
    _____
    _____
    _____

    // (6) p와 show()를 이용하여 colors 배열의 모든 객체의 색 출력. for 문 이용
    _____
    _____
    _____
}
```

```
255 0 0
255 0 0
0 255 0
0 0 255
```

4. 다음과 같은 Sample 클래스가 있다.

```
class Sample {  
    int *p;  
    int size;  
public:  
    Sample(int n) { // 생성자  
        size = n; p = new int [n]; // n개 정수 배열의 동적 생성  
    }  
    void read(); // 동적 할당받은 정수 배열 p에 사용자로부터 정수를 입력 받음  
    void write(); // 정수 배열을 화면에 출력  
    int big(); // 정수 배열에서 가장 큰 수 리턴  
    ~Sample(); // 소멸자  
};
```

다음 main() 함수가 실행되도록 Sample 클래스를 완성하라. **난이도 4**

```
int main() {  
    Sample s(10); // 10개 정수 배열을 가진 Sample 객체 생성  
    s.read(); // 키보드에서 정수 배열 읽기  
    s.write(); // 정수 배열 출력  
    cout << "가장 큰 수는 " << s.big() << endl; // 가장 큰 수 출력  
}
```

```
100 4 -2 9 55 300 44 38 99 -500  
100 4 -2 9 55 300 44 38 99 -500  
가장 큰 수는 300
```


7. 다음과 같이 원을 추상화한 Circle 클래스가 있다. Circle 클래스와 main() 함수를 작성하고 3개의 Circle 객체를 가진 배열을 선언하고, 반지름 값을 입력받고 면적이 100보다 큰 원의 개수를 출력하는 프로그램을 완성하라. Circle 클래스도 완성하라. **난이도 4**

```
class Circle {  
    int radius; // 원의 반지름 값  
public:  
    void setRadius(int radius); // 반지름을 설정한다.  
    double getArea(); // 면적을 리턴한다.  
};
```

```
원 1의 반지름 >> 5  
원 2의 반지름 >> 6  
원 3의 반지름 >> 7  
면적이 100보다 큰 원은 2개 입니다
```