A-1

Report

제목 : C++프로그래밍

과제 5번

과제내용

244-245p 연습문제 1, 2, 4번

246~248p 프로그래밍 과제 1, 3, 5번

미수행과제내역

담당교수 : 배성현 교수님

학과명 : 생명과학과

학번 : 5355002

이름 : 정유진

**일반 연습문제 과제**

244p

1. 다음 클래스 Point는 2차원 공간에서 한 점을 나타내는 클래스이다.

class Point {

private:

int x, y;

};

① 생성자와 소멸자를 클래스 내부에 정의하여 보라. 생성자는 중어진 x, y 좌표를 멤버 변수 x와 y에 저장한다.

Point(int n1, int n2) { // 생성자

x = n1;

y = n2;

}

~Point() { // 소멸자

delete n1;

delete n2;

}

② 생성자와 소멸자를 클래스 외부에 정의하여 보라.

Point::Point(int n1, int n2) { // 생성자의 외부정의

x = n1;

y = n2;

}

Point::~Point(){ // 소멸자의 외부정의

delete n1;

delete n2;

}

③ 생성자에서 초기화 리스트를 사용하여서 멤버 변수를 초기화하여 보라.

Point(int n1, n2) : x{n1}, y{n2} { } // 왼쪽은 멤버변수, 오른쪽은 매개변수

④ 생성자에서 디폴트 인수를 사용하여서 x, y값이 주어지지 않으면 0으로 초기화하라.

Point(){ // 인수가 전달되지 않았을 시에 x와 y의값을 0으로 초기화한다.

x = 0;

y = 0;

}

⑤ Point 클래스의 객체를 생성하고 (100, 200)으로 초기화하는 문장을 작성하여 보자.

Point obj; // 객체선언

obj(100, 200); // 인수를 전달하여 x는 100, y는 200으로 초기화

2. 사람을 나타내는 Person이라는 클래스는 다음과 같이 정의된다.

class Person

{  
private:

string name;

int snumber;

int age;

}

① 생성자와 소멸자를 클래스 내부에 정의하여 보라. 생성자는 다음과 같은 시그니처를 가진다.

Person(){ // 인수가 전달되지않았을 때 0으로 초기화

name = null;

snumber = 0;

age = 0;

}

Person(string n, int sn, int a){ // 인수가 전달되었을 때 각각의 인수로 초기화

name = n;

snumber = sn;

age = a;

}

② 초기화 리스트를 사용하여서 멤버 변수를 초기화하여 보라.

Person(string name, int snum, int age) : name{name}, snumber{snum}, age{age} { }

245p

4. Dog이라는 이름의 클래스를 설계한다. Dog 클래스는 다음과 같은 데이터를 가져야한다.

name: 강아지의 이름, 전용 멤버

breed: 강아지의 종류, 예를 들면 “요크셔테리어”, “푸들”, “말티즈” 등등, 공용 멤버

age: 강아지의 나이, 전용 멤버

Dog 클래스는 다음과 같은 생성자를 가져야 한다. 초기화되지 않은 데이터는 0으로 초기화하라.

Dog(string name, int age): 강아지의 이름과 나이로 초기화

Dog(string name, string breed, int age): 강아지의 이름, 종류, 나이로 초기화

class Dog{  
 private: string name; // 전용멤버

public: string breed; // 공용멤버

private: int age; // 전용멤버

Dog(){ // 인수가 전달되지않았을시에 0으로 초기화

name = 0;

breed = 0;

age = 0;

}

Dog(string n, int b){ // n과 b의 인수만 전달시에 실행되는 생성자

name = n;

breed = b;

age = 0;

}

Dog(string n, string b, int a){ // 모든 인수 전달시에 실행되는 생성자

name = n;

breed = b;

age = a;

}

}

**프로그래밍 과제**

246p

1. 앞장의 연습문제에서 책을 나타내는 Book 클래스를 정의한 바 있다. 생성자, 설정자, 접근자 등의 개념을 적용하여 Book 클래스를 다시 작성해보자. Book 클래스는 제목(title)과 저자(author)를 나타내는 멤버 변수를 가진다. 각 멤버 변수에 대하여 접근자와 설정자 함수를 정의한다. 생성자를 통하여 “Great C++”, “Bob”객체를 생성해본다.

1) 소스코드

#include <iostream>

#include <string>

using namespace std;

class Book {

private:

string title; //제목

string author; // 저자

public:

Book(string t, string a) { // 생성자

title = t;

author = a;

}

string getTitle() { return title; }

string getAuthor() { return author; }

};

int main() {

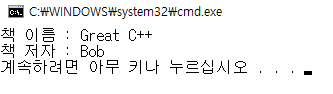
Book obj("Great C++", "Bob"); // 객체 선언 및 초기화

cout << "책 이름 : " << obj.getTitle() << endl;

cout << "책 저자 : " << obj.getAuthor() << endl;

}

2) 출력결과



3) 소스코드 설명

책이름과 책저자를 출력해주는 프로그램입니다. 생성자를 통하여 초기화를하고 접근자를 통하여 리턴값을 받아 책이름과 책저자를 출력합니다.

247p

3. 상자를 나타내는 Box라는 이름의 클래스를 설계하라. Box 클래스는 상자의 길이(length), 너비(width), 높이(height) 등의 변수를 가진다.

1) 소스코드

#include <iostream>

#include <string>

using namespace std;

class Box {

private:

int length; // 길이

int width; // 넓이

int height; // 높이

public :

Box() {

length = 0;

width = 0;

height = 0;

}

Box(int l, int w, int h) : length{l}, width{w}, height{h} {}

int getLength () { return length; } //접근자

int getWidth () { return width; }

int getHeight () { return height; }

void setLength(int l) { length = l; } // 설정자

void setWidth(int w) { width = w; }

void setHeight(int h) { height = h; }

int getVolume() { return length \* width \* height; } // 부피

void empty();

void print();

};

void Box::empty(){

if (length \* width \* height == 0) // 부피가 0일때 빈상자

cout << "박스가 비어있습니다." << endl;

}

void Box::print() { // 상자의 정보 출력

cout << "상자의 길이 :" << length << endl;

cout << "상자의 너비 : " << width << endl;

cout << "상자의 높이 : " << height << endl;

cout << "상자의 부피 : " << getVolume() << endl;

}

int main() {

Box obj1; // 매개변수가 없는 생성자 호출

Box obj2(3, 2, 4); // 매개변수가 있는 생성자 호출

Box obj3;

obj3.setHeight(3); // 설정자를 이용한 초기화

obj3.setLength(2);

obj3.setWidth(5);

cout << "상자 #1" << endl;

obj1.print();

obj1.empty(); // 빈상자확인

cout << "상자 #2" << endl;

obj2.print();

obj2.empty();

cout << "상자 #3" << endl;

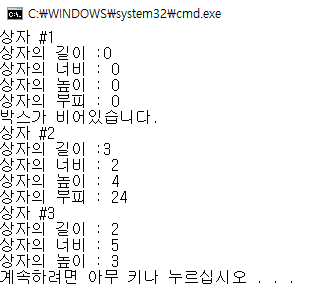
cout << "상자의 길이 : " << obj3.getLength() << endl;

cout << "상자의 너비 : " << obj3.getWidth() << endl;

cout << "상자의 높이 : " << obj3.getHeight() << endl;

obj3.empty();

}2) 출력결과



3) 소스코드 설명

상자의 정보와 부피를 구하여 출력해주는 프로그램입니다. 생성자를 2개 생성하고 각각의 경우를 출력하여 확인하였습니다. 부피가 0일경우에 박스가 비어있다는 문장을 출력해줍니다. 생성자와 함수를 호출하지 않아도 접근자와 설정자를 이용하여 값을 초기화하고 받아올 수 있습니다.

248p

5. 앞장의 연습문제에서 우리는 복소수를 나타내는 Complex 클래스를 작성한 바 있다. 복소수는 실수부와 허수부로 이루어진다. 이번 장에서 학습한 생성자, 접근자, 설정자를 이용하여 Complex 클래스를 다시 작성하라. 특히 이번 장에서는 복소수의 덧셈을 구현하는 add() 함수를 작성하고 테스트하자. add()는 다음과 같은 원형을 가진다.

Complex add(Complex a, Complex b);

1) 소스코드

#include <iostream>

using namespace std;

class Complex{

double r; // 실수

double i; // 복소수

public:

Complex() { // 인수 미전달시에 실행되는 생성자

r = 0;

i = 0;

}

Complex(double \_r, double \_i) { // 인수전달시에 실행되는 생성자

r = \_r;

i = \_i;

}

double getR() { return r; } // 접근자

double getI() { return i; }

void setR(double \_r) { r = \_r; } // 설정자

void setI(double \_i) { i = \_i; }

void print();

};

void Complex::print() {

if (i >= 0)

cout << "( " << r << " + " << i << "i)"; // i가0보다클때

else

cout << "( " << r << " - " << i\*-1 << "i)"; // i가 0보다작을때

}

Complex add(Complex a, Complex b);

int main() {

Complex obj1(5, 3), obj2(3, -4);

Complex result;

result = add(obj1, obj2); // add함수의 반환값을 받음

obj1.print();

cout << " + ";

obj2.print();

cout << " = ";

result.print();

}

Complex add(Complex a, Complex b) {

Complex num;

double r, i;

r = a.getR() + b.getR(); // a와b의 r값을 더한 값을 r에저장

num.setR(r); // 설정자를 이용하여 객체 num의 r값을 초기화

i = a.getI() + b.getI(); // a와b의 i값을 더한 값을 i에저장

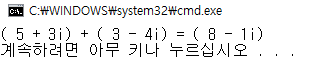
num.setI(i); // 설정자를 이용하여 객체 num의 i값을 초기화

return num; // class Complex의 객체 num을 반환

cout << endl;

}

2) 출력결과



3) 소스코드 설명

복소수들의 합을 출력하여주는 프로그램입니다. Complex 클래스가 자료형인 add 함수를 선언하여 복소수의 합을 구현하였습니다. 객체 3개를 선언하고 생성자를 통하여 초기화된 2개의 객체 obj1, obj2를 add함수에 전달하여 리턴받은 값을 객체 result에 저장하여 복소수의 합을 출력하였습니다.

4) 의견 및 고찰

생성자와 설정자를 배우면서 생성자가있는데 설정자를 어디에 쓰는가 의문이 들었는데 직접 코딩을 하고나서 설정자의 필요성에대해 알게되었다. 아직 초기화리스트라는 단어가 익숙하지않아서 앞으로도 코딩을 하면서 꾸준히 사용해봐야 될 듯 하다.

5번 문제의 경우에는 Chapter 4에서 풀어본 문제 8번에서 좀더 기능을 추가한 프로그램이다. 당시에는 더하기를 출력해주는 print\_plus함수와 빼기를 출력해주는 print\_mius 함수를 통하여 복소수의 +와 -를 출력하였지만 이번에 한번더 문제를 코딩하는 것을 고민해보면서 하나의 print함수를 사용하여 if-else문을 이용하여 허수부분에서 미만의 값을 전달받았을 시에 표시해주는 부호를 -로 변경하고 0미만의 값에 -1를 곱하여 자연수로 출력이 되게 하였다.