

=====

## 고급프로그래밍1 중간고사- 코딩

학과 : 학번 : 이름 : 점수 ( )

=====

### 문제

1. (배점 : 10점) 두자리 자연수 계산과정 프로그램
2. (배점 : 10점) 정렬 프로그램
3. (배점 : 15점) 여러가지 생성자를 만들어 보는 프로그램
4. (배점 : 15점) 좌표연산 객체 지향 프로그램

### 제출방법

- 작성한 코딩답지.hwp를 사이버 캠퍼스에 과제 --> 중간고사 (코딩)에 업로드하여 제출
  - \* 애플의 맥킨토시나 iMac 사용자는 코딩답지.txt를 이용할 것.
- 시험시간 종료 5분전과 2분전에 반드시 그때까지의 결과를 임시로 upload 하고, 최종제출 시에 최종제출물로 바꾸어서 제출할 것.
  - \* 왜냐하면, 시간이 종료되면 upload 안되기 때문.

1. (배점 : 10점) 두자리 자연수 계산과정 프로그램

가. 문제 요구사항 :

- 곱하기 과정을 보여주기 위한 상호작용 프로그램의 작성
- 두 개의 2자리의 자연수를 입력받는다.
  - \* 첫 번째 자연수 변수 : num1
  - \* 두 번째 자연수 변수 : num2
- 곱셈 계산과정을 출력한다.
  - \* 예)

12 * 34 계산과정	98 * 76 계산과정
$  \begin{array}{r}  12 \\  * 34 \\  \hline  48 \\  36 \phantom{0} \\  \hline  = 408  \end{array}  $	$  \begin{array}{r}  98 \\  * 76 \\  \hline  588 \\  686 \phantom{0} \\  \hline  = 7448  \end{array}  $

나. 힌트

- %연산자와 /연산자를 이용하라.

다. 프로그램 결과의 확인방법: 아래와 같이 98과 76을 입력하였을 경우, 결과가 다음과 같이 나오면 됨

```

C:\Users\T2\Desktop\2019년1...
*****
두자리 자연수의 곱 계산 프로그램
*****
2자리 자연수를 입력하시오 : 98
2자리 자연수를 하나 더 입력하시오 : 76

      98
   x   76
   -----
(98) x 6      is 588
(98) x 70     is 6860
   -----
      = 7448
*****
    
```

라. 이용할 소스코드의 다른 부분은 건드리지 말고, 성할 것.

// 작성할 부분

부분만 입력하여 완

마. 이용할 코드

(반드시 다음의 소스코드를 copy하여서 이용할 것)

```
#include<stdio.h>

int main()
{
    int num1, num2;

    printf("\n*****");
    printf("\n 두자리 자연수의 곱 계산 프로그램 ");
    printf("\n*****");
    printf("\n 2자리 자연수를 입력하시오 : ");
    scanf("%d",&num1);
    printf(" 2자리 자연수를 하나 더 입력하시오 : ");
    scanf("%d",&num2);
    printf("\n\t\t\t %4d",num1);
    printf("\n\t\t\t %4d",num2);
    printf("\n*****");
    // 작성할 부분

    printf("\n*****");
    printf("\n\t\t\t %4d", num1 * num2 );
    printf("\n*****");

    return 0;
}
```

2. (배점 : 10점) 정렬 프로그램

가. 요구 사항 :

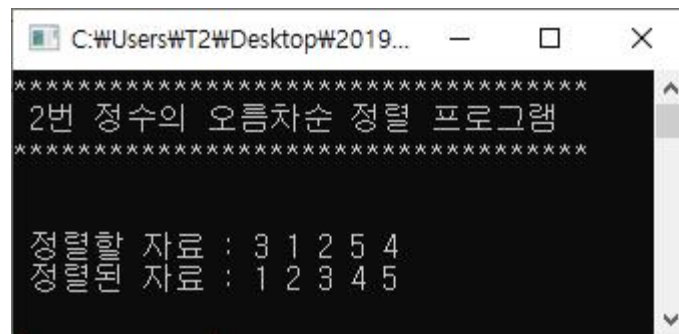
- 자료를 call by reference로 정렬하는 exchange() 함수를 작성하라.
- 그리고, 정렬프로그램을 호출하는 부분도 완성하라

나. 힌트 : **파란색 작성부분만 추가하여 완성** (소스코드의 다른 부분을 건드리지 말 것).

- exchange 함수의 원형은 다음과 같고, main함수의 5번째 줄에 원형이 선언되어 있다.  
`void exchange(int * , int *);`

다. 프로그램 결과의 확인방법:

- 잘 작성되었다면 다음과 같은 결과가 출력된다.



```
*****
2번 정수의 오름차순 정렬 프로그램
*****
정렬할 자료 : 3 1 2 5 4
정렬된 자료 : 1 2 3 4 5
```

라. 이용할 소스코드의 다른 부분은 건드리지 말고,  
성할 것.

**// 작성할 부분**

부분만 입력하여 완

마. 이용할 코드

(반드시 다음의 소스코드를 copy하여서 이용할 것)

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

/* 2019년 1학기 고급프로그래밍 1 중간고사 (코딩) */
/* 문제 2. 정수의 오름차순 정렬 프로그램 */

int main()
{
    const int N = 5;
    int i, j;
    int A[5] = { 3, 1, 2, 5, 4};

    void exchange(int * , int *) ;    //exchange 함수 원형선언

    cout << "*****" << endl;
    cout << " 2번 정수의 오름차순 정렬 프로그램 " << endl;
    cout << "***** \n \n" << endl;

    cout << " 정렬할 자료 : ";
    for (i=0; i<N; i++)
        cout << A[i] << " " ;
    cout << endl;

    for (i=0; i<N; i++) {
        for (j=0; j<N-1; j++) {
            if(A[i] < A[j]) {
                // 작성할 부분 : exchange 함수 호출부분
            }
        }
    }

    cout << " 정렬된 자료 : ";
    for(i=0; i<N; i++)
        cout << A[i] << " " ;
    cout << endl;

    return 0;
}

void exchange(int *a, int *b)
{
    int temp;
    // a와 b의 값을 temp를 이용하여 교환
    // 작성할 부분
}
```

3. (배점 : 15점) 여러가지 생성자를 만들어 보는 프로그램

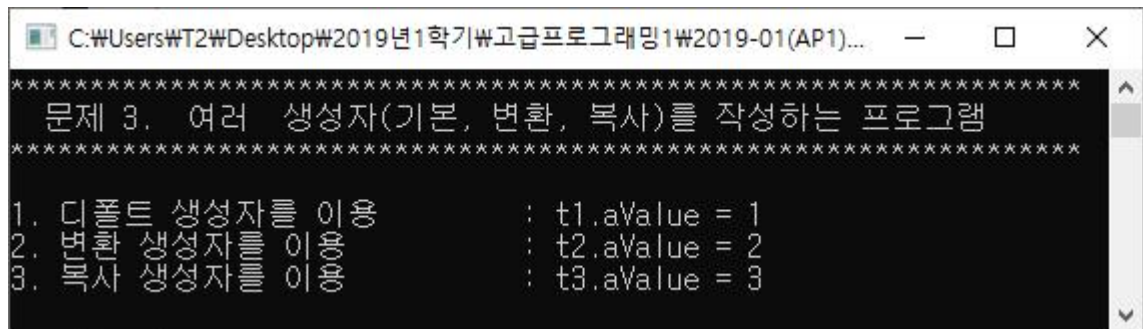
가. 요구 사항 :

- 디폴트 생성자
  - \* 멤버 데이터(m\_Value)의 값을 1로 초기화할 것.
- 변환 생성자
  - \* 멤버 데이터(m\_Value)의 값을 매개변수(파라미터)로 초기화할 것.
- 복사 생성자
  - \* 복사할 객체의 멤버 데이터(m\_Value) 값으로 m\_Value 값을 설정할 것.

나. 힌트 : **파란색 생성자 부분만 작성·수정하여 완성** (소스코드의 다른 부분을 건드리지 말 것).

다. 프로그램 결과의 확인방법:

- 3개의 생성자를 잘 작성하였다면, 아래 그림의 1. ~ 3.과 같이 (1, 2, 3)이 각각 출력됨



```
*****
문제 3. 여러 생성자(기본, 변환, 복사)를 작성하는 프로그램
*****
1. 디폴트 생성자를 이용      : t1.aValue = 1
2. 변환 생성자를 이용       : t2.aValue = 2
3. 복사 생성자를 이용       : t3.aValue = 3
*****
```

라. 이용할 소스코드의 다른 부분은 건드리지 말고,

**// 작성할 부분**

부분만 입력하여 완

성할 것.

마. 이용할 코드

(반드시 다음의 소스코드를 copy하여서 이용할 것)

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

/* 2019년 1학기 고급프로그래밍 1 중간고사 (코딩) */
/* 문제 3 여러 생성자 작성 프로그램 */
class CTest1
{
public:
    CTest1()
    {
        // 기본(디폴트) 생성자 작성할 부분
    }
    CTest1(int nParam)
    {
        // 변환 생성자 작성할 부분
    }
    CTest1 (const CTest1 &obj )
    {
        // 복사 생성자 작성할 부분
    }
    int getValue(void)
    {
        return aValue;
    }
    void setValue(int nParam)
    {
        this->aValue = nParam;
    }
private:
    int aValue;
};

int main()
{
    cout << "*****" << endl;
    cout << " 문제 3. 여러 생성자(기본, 변환, 복사)를 작성하는 프로그램" << endl;
    cout << "*****" << endl;
    cout << endl;

    CTest1 t1; // 디폴트 생성자 호출
    cout << "1. 디폴트 생성자를 이용 \t: t1.aValue = " << t1.getValue() << endl;

    CTest1 t2(2); // 변환 생성자 호출
    cout << "2. 변환 생성자를 이용 \t\t: t2.aValue = " << t2.getValue() << endl;

    t1.setValue(3);
    CTest1 t3(t1); // 복사 생성자 호출
    cout << "3. 복사 생성자를 이용 \t\t: t3.aValue = " << t3.getValue() << endl;
    return 0;
}
```

4. (배점 : 15점) 좌표연산 객체 지향 프로그램

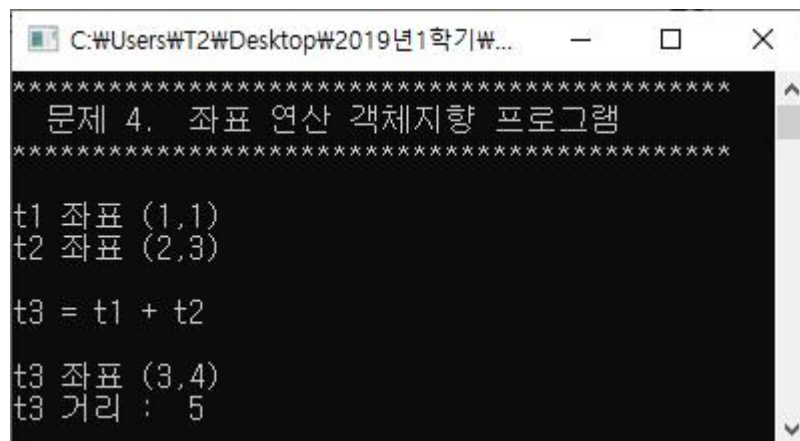
가. 문제 요구사항 :

- 좌표 t3 = t1과 t2 객체를 이용하여 생성하라.
  - \*  $t3 = t1 + t2$
  - \* 생성자를 만들어라.
  - \* 연산자함수 (+, =)을 만들고 overload 하라.

나. 힌트

- cCord 클래스는 멤버변수로 xVal과 yVal을 갖는다.
  - \* 멤버변수의 접근제어 지시자는 private임
- cCord 클래스는 멤버 메소드로 print()와 distance를 갖는다.
  - \* print()는 좌표를 출력하는 것임
  - \* distance()는 좌표의 거리를 계산하는 것임.

다. 프로그램 결과의 확인방법: 실행시킨 후 다음의 결과가 다음과 같이 나오면 됨



```
*****
문제 4. 좌표 연산 객체지향 프로그램
*****

t1 좌표 (1,1)
t2 좌표 (2,3)

t3 = t1 + t2

t3 좌표 (3,4)
t3 거리 : 5
```

라. 이용할 소스코드의 다른 부분은 건드리지 말고,  
하여 완성할 것.

// 작성할 부분 1 ~ 3

부분만 입력



마. 이용할 코드

(반드시 다음의 소스코드를 copy하여서 이용할 것)

```
#include<iostream>
#include<cmath>

using namespace std;

/* 2019년 1학기 고급프로그래밍 1 중간고사 (코딩) */
/* 문제 4 좌표 연산 객체지향 프로그램 */

class CCord
{
public:
    CCord(int nParam1, int nParam2)
    {
        // 코드 작성할 부분1
    }
    CCord& operator=(CCord &rhs)
    {
        // 코드 작성할 부분2
    }
    CCord operator+(CCord rhs)
    {
        // 코드 작성할 부분3
    }
    void print(string nParam)
    {
        cout << nParam << " 좌표 (" << xVal << ", " << yVal << ")" << endl;
    }
    void distance(string nParam)
    {
        cout << nParam << " 거리 : " << sqrt(xVal*xVal + yVal*yVal) << endl;
    }

private:
    double xVal;
    double yVal;
};

int main()
{
    cout << "*****" << endl;
    cout << " 문제 4. 좌표 연산 객체지향 프로그램" << endl;
    cout << "*****" << endl;
    cout << endl;

    CCord t1 (1,1);
    CCord t2 (2,3);
    t1.print("t1");
    t2.print("t2");

    CCord t3 = t1 + t2;

    cout << endl << "t3 = t1 + t2 " << endl << endl;

    t3.print("t3");
    t3.distance("t3");

    return 0;
}
```