

# 프로그래밍 언어 (2) 실습

10주차

# C++ 심화 문제

- ex1)

>> 아래와 같은 특징을 가지는 Car클래스를 작성하시오.

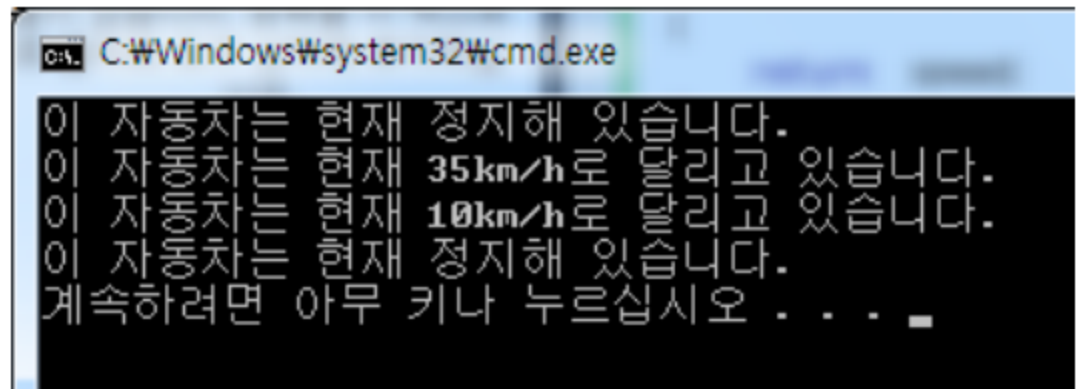
- int형 변수 speed를 가져야 하며 외부에서 접근이 불가능하도록 한다.
- 생성자는 int형 값을 하나만 받아서 speed변수를 초기화해주도록 한다.(기본생성자는 X)
- int형 값을 입력으로 받아 스피드를 올리는 speedUp과 스피드를 낮추는 speedDown 멤버함수를 작성한다.
  - 두 함수 모두 입력으로 받는 스피드값은 0보다 크고 30보다 작아야하며, 최대스피드는 200, 최하 스피드는 0이 되도록 작성하시오.
- 현재 speed값을 반환하는 currentSpeed멤버함수를 작성하시오.
- 출력 연산자를 오버로딩하여 cout을 이용해 현재 스피드를 출력할 수 있도록 하시오.

# C++ 심화 문제

- ex1) Result

## main 함수와 출력화면

```
int main()  
{  
    Car car1(0);  
    cout << car1;  
    car1.speedUp(10);  
    car1.speedUp(25);  
    cout << car1;  
    car1.speedDown(25);  
    cout << car1;  
    car1.speedDown(10);  
    cout << car1;  
  
    return 0;  
}
```



```
C:\Windows\system32\cmd.exe  
이 자동차는 현재 정지해 있습니다.  
이 자동차는 현재 35km/h로 달리고 있습니다.  
이 자동차는 현재 10km/h로 달리고 있습니다.  
이 자동차는 현재 정지해 있습니다.  
계속하려면 아무 키나 누르십시오 . . .
```

# C++ 심화 문제

- ex1) Solution

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

class Car
{
private:
    int speed;

public :
    Car(int _speed=0): speed( _speed){}
    ~Car( ){}

    void speedUp( int _speed=10);
    void speedDown( int _speed=10);
    int currentSpeed( ) const;

    friend ostream& operator << (ostream& os,const Car& car);
};

void Car::speedUp( int _speed)
{
    if( _speed > 0 && _speed < 30)
    {
        speed += _speed;
        if(speed > 200) speed = 200;
    }
}
```

# C++ 심화 문제

- ex1) Solution

```
//현재 속도
int Car::currentSpeed() const
{
    return speed;
}
ostream& operator << (ostream& os,const Car& car)
{
    if(car.currentSpeed() > 0)
    {
        os << "이 자동차는 현재 " << car.speed << "km/h" << "로 달리고 있습니다." << endl;
    }else{
        os << "이 자동차는 현재 정지해 있습니다." << endl;
    }
    return os;
}
```

# C++ 심화 문제

- ex2)

- (1) 년도를 입력받아 (년도-1)년까지의 총 일수를 리턴하는 TotalDays함수를 만드시오.  
(윤년은 366일, 윤년이 아닐경우 365일을 더해나가는 식으로 구한다.)
- (2) 함수 오버로딩을 이용하여 년도와 월을 입력받아 해당년도/(월-1)월까지의 총 일수를 리턴하는 TotalDays함수를 만드시오. (1번에서 작성한 TotalDays함수로 구한 일수에 CountOfDays함수로 월의 합을 구한다.)
- (3) 함수 오버로딩을 이용하여 년도와 월과 일을 입력받아 해당년도/월/일까지의 총 일수를 리턴하는 TotalDays함수를 만드시오. (2번에서 작성한 TotalDays함수로 구한 일수에 날짜를 더한다.)
- (4) 년/월/일을 입력받아 해당일의 요일을 리턴하는 DayOfTheWeek 함수를 만드시오. (요일은 1년1월1일이 월요일임을 이용하여 구하시오. 출력은 "SUNDAY", "MONDAY", "TUESDAY", "WENDSDAY", "THURSDAY", "FRIDAY", "SATURDAY"와 같은 문자열 형태로 출력하시오.)
- (5) 지금까지 작성한 함수를 이용하여 2000년 1월 1일이 1년 1월 1일로부터 몇일이 지났는지 화면에 출력하고 몇요일인지를 출력하시오.

# C++ 심화 문제

- ex2) Result

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

enum Dates{SUN, MON, TUE, WED, THU, FRI, SAT};

bool isLeapYear(int year);
int CountsOfDay(int year, int month);
int TotalDays(int year);
int TotalDays(int year, int month);
int TotalDays(int year, int month, int day);
int DayEnumOfTheWeek(int year, int month, int day);
char* DayStringOfTheWeek(int year, int month, int day);

void main()
{
    cout << "1년 1월 1일은 월요일 입니다." << endl;
    int totaldays = TotalDays(2000,1,1);
    cout << "0001.01.01부터 " << "2000.01.01일까지 " << totaldays-1 << "일 경과." << endl;
    cout << "2000.01.01일은 " << DayStringOfTheWeek(2000,1,1) << "이다." << endl;
}
```

```
1년 1월 1일은 월요일 입니다.
0001.01.01부터 2000.01.01일까지 730119일 경과.
2000.01.01일은 SATURDAY이다.
계속하려면 아무 키나 누르십시오 . . .
```

# C++ 심화 문제

- ex2) Solution

```
//윤년
bool isLeapYear(int year)
{
    return ((year%4==0)&&(year%100!=0)) || (year%400!=0);
}
//월별날짜.
int CountsOfDay(int year,int month)
{
    int months[]={31,28,31,30,31,30,31,31,30,31,30,31};
    int leapMonths[]={31,29,31,30,31,30,31,31,30,31,30,31};

    return isLeapYear(year)?leapMonths[month-1]:months[month-1];
    //return isLeapYear(year)&&(month == 2)?29:months[month-1];
}
//1년부터 year-1년 까지 총 일수
int TotalDays(int year)
{
    int temp=0;
    for(int i=1;i < year;i++) //1년 기준
    {
        (isLeapYear(i)?temp+=366:temp+=365);
    }
    return temp;
}
```



# C++ 심화 문제

- ex2) Solution

```
//year년 1월부터 month-1월까지 총 일수
int TotalDays(int year, int month)
{
    int temp=TotalDays(year);
    for(int i=1; i < month; i++)
    {
        temp+=CountsOfDay(year, i);
    }
    return temp;
}

//year년 month월 day일까지 총 일수
int TotalDays(int year, int month, int day)
{
    int temp=TotalDays(year, month);
    temp+=day;
    return temp;
}

//요일 구하기.|
int DayEnumOfTheWeek(int year, int month, int day)
{
    int totaldays = TotalDays(year, month, day);
    return totaldays%7;
}
```

```
char* DayStringOfTheWeek(int year, int month, int day)
{
    switch(DayEnumOfTheWeek(year, month, day))
    {
        case SUN : return "SUNDAY";
        case MON : return "MONDAY";
        case TUE : return "TUESDAY";
        case WED : return "WEDNESDAY";
        case THU : return "THURSDAY";
        case FRI : return "FRIDAY";
        case SAT : return "SATURDAY";
    }
    return "UNKNOWN";
}
```

# M 기업 입사 문제

- ex3)

다음의 소스 코드는 에러를 포함하고 있다.

단 "한 문자" 만 바꾸어서(추가/변경만 가능) 프로그램이 정확히 20개의 "\*" 기호를 출력하도록 고쳐라.

답은 세 가지가 있다.

```
#include <stdio.h>

int main()
{
    int i, n = 20;
    for (i = 0; i < n; i--)
        printf("*");
    return 0;
}
```

# M 기업 입사 문제

- ex3) Solution

## 첫번째 방법

```
int main ( )
{
    int i,n=20;
    for(i=0;-i<n;i--) // i를 음수 부호를 붙여서 음수로 향하는 i를
        printf("*"); // 양수화 한다. 20 이 되는 순간 거짓이 되어
    return 0; // 탈출한다.
}
```

## 두번째 방법

```
int main ( )
{
    int i,n=20;
    for(i=0;i+n;i--) // i는 음수화 되어가고 n은 20에 고정 되어
        printf("*"); // 있으므로 더하여 0 이 되는 순간 탈출한다.
    return 0;
}
```

# M 기업 입사 문제

- ex3) Solution

## 세번째 방법

```
int main ( )
{
    int i,n=20;
    for(i=0;i<n;n--) // i를 n으로 바꾸어서 i를 고정 시킨후 n을
        printf("*"); // 감소 시켜 거짓으로 판단하고 탈출한다.
    return 0;
}
```

## 네번째 방법

```
int main ( )
{
    int i,n=20;
    for(i=0;~i<n;i--) // i를 비트 반전 시켜서 거짓을 만들어서
        printf("*"); // 탈출 시킨다.
    return 0;
}
```

Q & A