

## LAB #08 (05/02 목)

1. 아래의 코드를 기반으로 Polygon class을 상속받는 Rectangle class를 선언하고 둘레와 넓이를 구하는 프로그램을 작성하라. 단, 정사각형이라 가정한다.

```
int main()
{
    Polygon pol;
    Rectangle rec(4,10);

    cout << "--- Polygon class ---" << endl;
    pol.calcGirth();
    pol.calcArea();
    cout << "--- Rectangle class ---" << endl;
    rec.calcGirth();
    rec.calcArea();

    return 0;
}
```

### [참조 1]

```
class Polygon {
public:
    Polygon();
    Polygon(int point, float length);
    ~Polygon();

    virtual void calcGirth();
    virtual void calcArea();

protected:
    int mPoint; // 꼭지점 갯수
    double mLength; // 한 변의 길이
};
```

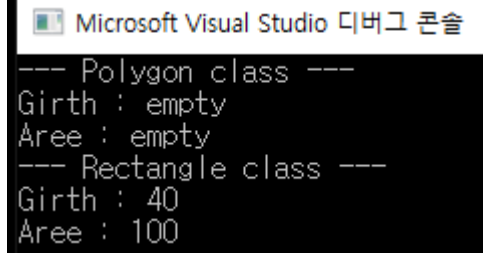
### [참조 2]

```
class Rectangle : public Polygon
{
public:
    Rectangle() {};
    Rectangle(int point, float length) : Polygon(point, length);
    ~Rectangle() {};

    void calcGirth() override;

    void calcArea() override;
};
```

1 – 출력화면 :



```
Microsoft Visual Studio 디버그 콘솔
--- Polygon class ---
Girth : empty
Area : empty
--- Rectangle class ---
Girth : 40
Area : 100
```

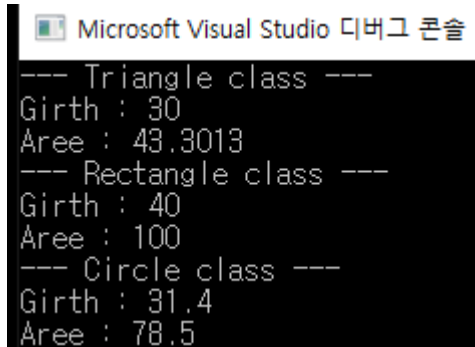
2. 1번 문제에서 작성한 프로그램을 기반으로 Polygon class를 상속하는 Triangle, Rectangle, Circle class를 선언하고 둘레와 넓이를 구하는 프로그램을 작성하라. 단, 모든 도형은 정다각형이라 가정한다.

```
int main()
{
    Triangle tri(3, 10);
    Rectangle rec(4, 10);
    Circle cir(0, 5);

    cout << "--- Triangle class ---" << endl;
    tri.calcGirth();
    tri.calcArea();
    cout << "--- Rectangle class ---" << endl;
    rec.calcGirth();
    rec.calcArea();
    cout << "--- Circle class ---" << endl;
    cir.calcGirth();
    cir.calcArea();

    return 0;
}
```

2 – 출력화면 :



```
Microsoft Visual Studio 디버그 콘솔
--- Triangle class ---
Girth : 30
Area : 43.3013
--- Rectangle class ---
Girth : 40
Area : 100
--- Circle class ---
Girth : 31.4
Area : 78.5
```

3. Ktx 열차가 출발역, 종착역을 포함하여 총 5개의 역에 정차한다. 이 때 각 역에서는 입력된 숫자만큼의 사람이 내리거나 탑승한다. 이 기차는 아래와 같은 조건을 만족하면서 운행된다고 가정한다.

1. 기차는 역 번호 순서대로 운행한다.
2. 출발역에서 내린 사람 수와 종착역에서 탄 사람 수는 0이다.

3. 각 역에서 현재 기차에 있는 사람보다 더 많은 사람이 내리는 경우는 없다.
  4. 기차의 정원은 최대 300명이고, 정원을 초과하여 타는 경우는 없다.
- 5개의 역에 대해 기차에서 내린 사람 수와 탄 사람 수가 주어졌을 때, 기차에 사람이 가장 많을 때의 사람 수를 계산하는 프로그램을 작성하라.

```
int main()
{
    Ktx k;

    // ???

    return 0;
}
```

#### [참조 1]

```
class Train {
public:
    Train() {}
    Train(int people)
    {
        mPeople = people;
    }
    ~Train() {}

    virtual int station(int takeOff, int takeOn);

protected:
    int mPeople; // 사람 수
};
```

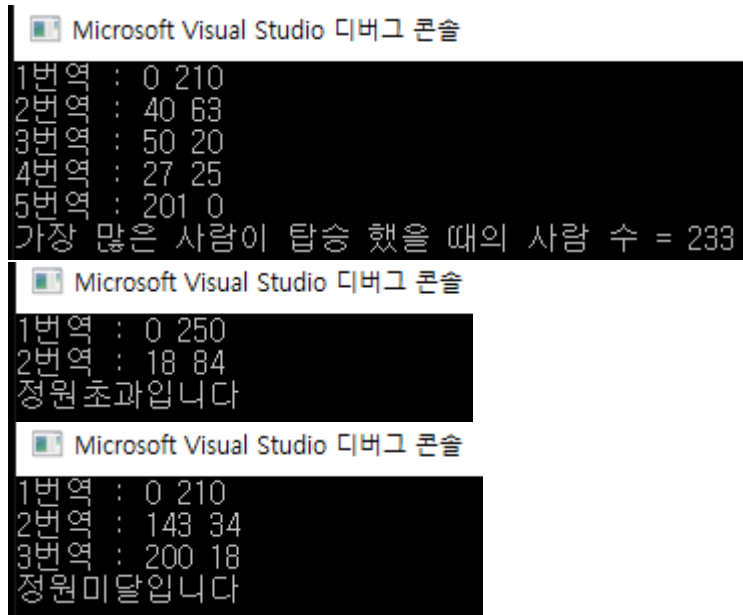
#### [참조 2]

```
class Ktx : public Train
{
public:
    Ktx() : Train(0) {}
    Ktx(int people) : Train(people)
    {}
    ~Ktx() {}

    // 기차에 사람이 타고 내리는 함수
    int station(int takeOff, int takeOn);

    int getPeople();
};
```

3 – 출력화면 :



4. 어벤져스 캐릭터 배틀 프로그램을 만들려고 한다. 3번 문제와 유사한 방식으로 Avengers 클래스를 상속받아 Character 클래스를 구현하고, 다음 조건에 맞게 구현하라. 사용자는 캐릭터를 선택하고 상대방 캐릭터는 랜덤으로 선택된다.

1. 각 캐릭터는 캐릭터 이름, 공격력, 방어력, 체력을 갖는다.
2. 공격하는 함수는 상대방에게 공격력을 가하고, 공격받는 함수(방어 함수)는 상대방의 공격력에서 자신의 방어력의 차만큼 자신의 체력을 감소한다.
3. 서로 공격을 주고 받다가, 어느 한쪽의 체력이 0이하가 되면 배틀을 종료한다. 공격을 주고 받을 때 마다 자신과 상대방의 캐릭터 체력을 출력하고 선공은 자신의 캐릭터가 먼저

	IronMan	CaptainAmerica	Thor
공격력	70	60	80
방어력	40	50	30
체력	100	100	100

하는 것으로 설정한다.

[참조 1]

```
class Avengers
{
public:
    Avengers() {
        name = "";
        attack_point = 0;
        defense_point = 0;
        health = 0;
    }
}
```

```

~Avengers() {}
//캐릭터 설정 함수
virtual void set(string _name, int _attack, int _defense, int _health)
{
}
//공격 함수
virtual int attack()
{
    return 0;
}
//방어 함수
virtual void defense(int _attack_point)
{
}
//캐릭터 정보 출력 함수
virtual void print_info()
{
}
protected:
    string name;//캐릭터 이름
    int attack_point;//공격력
    int defense_point;//방어력
    int health;//체력
};

```

[참조 2]

```

class Character: public Avengers
{
public:
    //구현
    int get_health() { return health; }
};

```

[참조 3]

```

int main()
{
    Character my_char;
    Character enemy_char;
    /*
    구현
    */

    cout << endl << "--Battle--" << endl;
    cout << "My Life: " << my_char.get_health() << "Wt"
        << "Enemy Life:" << enemy_char.get_health() << endl;

    while (1)
    {
        /*
        구현
        */
    }
    return 0;
}

```

[출력화면 1]

```
Choose your character(IronMan, CaptainAmerica, Thor) : IronMan
--My Character--
Name : IronMan
Attack_Point : 70
Defense_Point : 40
Health : 100
--Enemy Character--
Name : CaptainAmerica
Attack_Point : 60
Defense_Point : 50
Health : 100

--Battle--
My Life: 100      Enemy Life:100
My Life: 100      Enemy Life:80
My Life: 80       Enemy Life:80
My Life: 80       Enemy Life:60
My Life: 60       Enemy Life:60
My Life: 60       Enemy Life:40
My Life: 40       Enemy Life:40
My Life: 40       Enemy Life:20
My Life: 20       Enemy Life:20
My Life: 20       Enemy Life:0
You Win!
```