

Programming Methodology

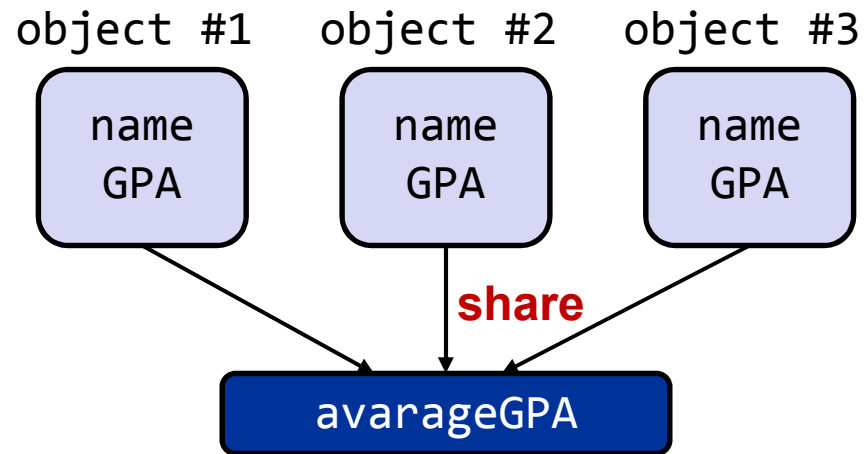
Practice Session #7

Static Variables, New, Delete

Static Variables (1)

- 특정 class type의 모든 객체들이 공유할 수 있는 변수.

```
class Student
{
    private:
        char name[30];
        double GPA;
        static double avarageGPA;
}
```



- 반드시 클래스 정의 외부에서 초기화시켜야 한다.
 - `double Student::avarageGPA = 0;`

Static Variables (2)

```
class ExampleClass
{
    private:
        static int var1;    // private static variable
    public:
        static int var2;    // public static variable
        void func() { cout << ++var1 << ", " << var2 << endl; }
};

int ExampleClass::var1 = 0; // var1 초기화
int ExampleClass::var2 = 0; // var2 초기화

void main()
{
    ExampleClass obj;
    obj.var2 = 5;           // public static variable에 접근 방법 #1
    obj.func();             // 1, 5
    ExampleClass::var2 = 3; // public static variable에 접근 방법 #2
    obj.func();             // 2, 3
}
```

New

- 데이터를 저장할 메모리 공간을 **런타임(runtime)에 확보**하기 위해 **new** 키워드를 사용한다.
 - 생성해야 할 데이터의 개수가 정확하지 않은 경우에 효율적으로 메모리를 활용할 수 있다.
- **new** 키워드는 객체를 메모리의 **heap area**에 저장하고 할당된 메모리 주소를 가리키는 **pointer**를 return한다.
 - Local area가 아닌 heap area에 생성되므로, **new**를 이용해 생성된 객체는 함수의 수행이 끝나도 사라지지 않는다.
 - **new**를 이용해 객체를 생성하면 constructor가 호출된다.
- **new** 키워드를 이용해 **array**를 생성할 수도 있다.

Delete

- **new**로 할당된 메모리는 반드시 **delete** 키워드를 이용해 해제시켜야 한다.
 - **delete**로 해제되지 않은 데이터는 **프로그램이 종료되어도 메모리를 계속 차지**하고 있어서 리소스의 낭비가 일어난다.
 - 오랫동안 동작하는 프로그램(ex. Server application)의 경우 특히 중요하다.
- **new**로 할당된 **array**를 삭제하기 위해서는 **delete []** 키워드를 사용한다.

New & Delete

```
class ExampleClass
{
    public:
        ExampleClass();           // #1, default constructor
        ExampleClass(int param); // #2
        ~ExampleClass();         // Destructor
};

void main()
{
    int * intArray = new int[5]; // int array 생성
    ExampleClass * obj = new ExampleClass(3); // 객체 생성, #2 호출
    ExampleClass * objArray = new ExampleClass[10]; // array 생성, #1 호출

    delete obj; // obj 객체 삭제, destructor 호출
    delete [] intArray; // intArray 삭제
    delete [] objArray; // objArray 내의 모든 객체의 destructor 호출
}
```

Sample Practice (1)

- **Salesman** Class를 구현한다.
 - Member Variables
 - `char*` name;
 - `static int` salesmanCount;
 - `static double` asset;
 - Member Functions
 - `Salesman(const char* name);` // Constructor
 - `~Salesman();` // Destructor
 - `void sell(int income);`
 - `void spend(int amount);`
 - `void report();`

Sample Practice (2)

- Member Variables
 - **char*** name
 - 세일즈맨의 이름.
 - **static int** salesmanCount
 - 프로그램 내에서 현재 생성되어 있는 객체의 수를 카운트하는 static variable. Salesman 객체가 **생성되거나(new)** 사라지면 **(delete)** 값이 변한다.
 - **static double** asset
 - 가지고 있는 총 자산. 세일즈맨이 **물건을 팔거나(sell)** 지출을 하면 **(spend)** 값이 변한다.

Sample Practice (3)

■ Member Functions

- Salesman(**const char*** name)

- 이름을 parameter로 받아 Salesman 객체를 생성할 때 호출되는 constructor. Default constructor를 따로 선언하지 않았으므로 모든 객체는 **const char*** type의 parameter와 함께 생성되어야 한다.
- 새로운 객체가 생성되므로 **static** variable인 **salesmanCount**의 값을 **1만큼 증가**시킨다.

- ~Salesman()

- Salesman object가 파괴될 때 호출되는 destructor. 객체가 파괴되므로 **salesmanCount**의 값을 **1만큼 감소**시킨다.

Sample Practice (4)

■ Member Functions

- **void** sell(**int** income)

- 세일즈맨이 물건을 팔고 **income** 만큼의 돈을 벌었을 때 사용한다. 총 자산이 증가하므로 **static variable**인 **asset**의 값을 증가시킨다.

- **void** spend(**int** amount)

- 세일즈맨이 **amount** 만큼의 자산을 사용했을 때 사용한다. 총 자산이 감소하므로 **static variable**인 **asset**의 값을 감소시킨다.
- 만약 **asset**이 **amount**보다 적다면 에러 메시지를 출력한다.

Sample Practice (5)

- Member variable들은 반드시 **private**으로 선언한다.
- 세일즈맨의 고용과 해고는 반드시 **new**와 **delete** 키워드를 이용한다.
- 세일즈맨의 고용과 해고 메시지는 각각 **constructor**와 **destructor**에서 출력하도록 한다.
- 세일즈맨의 수입과 지출 메시지는 각각 **sell()** 함수와 **spend()** 함수에서 출력하도록 한다.

Sample Practice (6)

```
Hire "Chris". Totally 1 salesmen were hired.
Hire "Scarlet". Totally 2 salesmen were hired.
Hire "Bruce". Totally 3 salesmen were hired.
    "Chris" spent $100. Total asset became $400.
    "Scarlet" spent $200. Total asset became $200.
    "Chris" earned $300. Total asset became $500.
    "Scarlet" spent $400. Total asset became $100.
    "Scarlet" earned $500. Total asset became $600.
    "Bruce" spent $600. Total asset became $0.
    "Bruce" tried to spend $700, but there's not enough money.

Fire "Bruce". Totally 2 salesmen remained.

Hire "Jennifer". Totally 3 salesmen were hired.
    "Jennifer" earned $2000. Total asset became $2000.
    "Scarlet" spent $1500. Total asset became $500.
Fire "Scarlet". Totally 2 salesmen remained.

Hire "Sarah". Totally 3 salesmen were hired.
    "Sarah" earned $500. Total asset became $1000.
    "Jennifer" tried to spend $2000, but there's not enough money.

Fire "Chris". Totally 2 salesmen remained.
Fire "Sarah". Totally 1 salesmen remained.
Fire "Jennifer". Totally 0 salesmen remained.
계속하려면 아무 키나 누르십시오 . . .
```