Lab 2

Introduction of Programming Environment

Basic of C++

input & output for C++

output : cout<<1<<'a'<<"String"<<endl;

input : cin >>val1>>val2;

Function Overloading

```
#include <iostream>

void function(void)
{
        std::cout << "function(void) call" << std::endl;
}

void function(int a, int b)
{
        std::cout << "function("<<a<<","<<b<<") call"<<std::endl;
}

int main(void)
{
        function();  func
        tion(12, 13);
        return 0;
}
```

- Writing multiple functions with SAME funct ion name but DIFFERENT parameters!

! Same name functions : Error occur on C

Default Parameter

```
#include <iostream>

using std::cout;
using std::endl;

int func(int a = 0)
{
        return a+1;
}

int main(void)
{
        cout << func(11) << endl;
        cout << func() << endl;
        return 0;
}
```

- With default parameter set, value of the param eter is automatically assigned as default value even though actual parameter is not put when function is called.

C++ Class

- Class라는 새로운 형 사용
  - 변수가 사용하는 메모리 크기 (Memory allocation)
  - 변수가 가질 수 있는 정보 (Data element)
  - 변수에게 가능한 조작 (Operations)
- Class의 정의
  - 변수들과 연관된 함수들을 결합시킨 새로운 형 Object with variables and operations
  - 클래스를 선언함으로써 새로운 형 생성 Class becomes new type
  - 구조체를 선언하는 것도 새로운 형을 생성하는 것이지만 함수까지 결 합 한다는 것이 구조체와 클래스의 차이 struct has no operations but Class DO

C++ Class

- 용어
  - Member variable:
    - 클래스 내의 변수
    - Variable defined inside Class
  - Member function or method:
    - 클래스 내의 함수, 객체가 무엇을 하는지를 결정
    - Function defined inside Class. Describes what the class does.
  - Object :
    - 클래스에 의해 만들어진 변수
    - Object is instantiation of Class

C++ Class

- 클래스 선언 Declaration of Class
- class 클래스 명

{

  Member variable;

  Member fuction;

};

```
Ex)
class Fishbread
{
        string content;
        void Wrapped();
};
```

클래스 선언 시 메모리 할당이 되는 것이 아니라 객체를 생성할 때 메모리 할당이 일어남.
Memory is not allocated when Class is declared but when instantiated (when instantiation is assigned to variable of type Class

C++ 붕어빵 class (Fishbread Class)

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

class Fishbread
{
public:
   Fishbread()
   {
      cout<<"create : "<<content<<endl;
   }
   Fishbread(int argCost, string argCtent)
   {
      cost=argCost;
      content=argCtent;
      cout<<"create : "<<content<<endl;
   }
   ~Fishbread()
   {
       cout<<"we finish eating the <"+content+"> fishbread"<<endl;
   }
   int getCost() { return cost;}
   void setCost(int c) {cost=c;}

private:
   int cost;
   string content;
};
```

```
int main()
{

Fishbread fish1(500,"red bean");
Fishbread *fish2=new Fishbread(300, "cream");
cout<<"how much? "<<fish1.getCost()<<endl;
fish2->setCost(1);
cout<<"how much? "<<fish2->getCost()<<endl;

delete (fish2);
return 0;

}
```

C++ Class

- 접근제어 지시자 Access modifier
  - public : 어디서든 접근 허용 can be accessed everywhere
  - protected : 상속관계에 놓여 있을 때, 유도 클래스에서의 접근 허용 can be accessed by friend, or derived class
  - private : 클래스 내(클래스 내에 정의된 함수)에서만 접근 허용 can be accessed only by the class
  - 명시되지 않은 경우에 대해서는 default로 public

- 클래스에서 private은 외부로 객체의 data를 마음대로 접근할 수 없도록 하기 위해(캡슐화) 사용하며, 캡슐화된 데이터에 접근하기 위해서는 public으로 선언된 메소드를 선언

Member variable (or data) is set private in order to block access from outside. In order to modify or read the data, public method will be used.

C++ Class - Example

```
class Fishbread
{
private:
    int cost;
    int seller;  string content;  int roasting;
public:
    int GetCost(){return cost;}
    void SetCost(int c) {cost=c;}
};
```

```
int main()
{

    Fishbread fish1;
    fish1.SetCost(500);
    cout<<"Price?"<<fish1.GetCost()<<endl;
    fish1.SetCost(700);
    cout<<"Price?"<<fish1.GetCost()<<endl;


}
```

C++ Basics & Data Types

- 클래스 메소드의 구현
리턴형 클래스명::함수명(매개변수1,매개변수2,..)
{
…
}
- How method is defined
returnType className::functionName (parameters)
{
…
}

```
class Fishbread
{
private:
    int cost;  int seller;
    string content;  int roasting;  public:
    int GetCost();
    void SetCost(int argCost);
};

int Fishbread::GetCost()
{
return cost;
}
void Fishbread::SetCost(int argCost)
{
cost=argCost;
}
```

C++ Basics & Data Types

- const 멤버 함수 (const member function)

  멤버 함수를 const 선언 시,
  해당 클래스의 모든 멤버 값 변경 불능
  When member function is declared con
  st,

```
class Fishbread
{
public:
    void SetCost() const;
private:
    int cost;
};
void Fishbread::SetCost() const
{
    cost=500; //error
}
```

C++ - header file

- 클래스 선언과 멤버 함수 작성
  - 클래스와 사용자 간의 통신 인터페이스
  - 클래스의 자료형, 함수 종류 알림
  - *.h 파일 사용

  Header file contains basic info of what the clas s looks like (overview of class)
- 함수 정의
  - 함수의 구체적 동작 정의
  - *.cpp 파일 사용

  For exact definition of the class in header file, we make cpp file and put definitions.

C Fishbread.h

```
#include <iostream>
#include <string>
using std::string;

class Fishbread
{
public:
    Fishbread();
    Fishbread(int argCost, string argCtent);
    ~Fishbread();
    int getCost();
    void setCost(int c);

private:
    int cost;
    string content;
};
```

C++ class body & main

```cpp
#include "Fishbread.h"
using std::cout;
using std::endl;


Fishbread::Fishbread(){}

Fishbread::Fishbread(int argCost, string argContent)
{
    cost=argCost;
    content=argContent;
}

Fishbread::~Fishbread()
{
    cout<<"we finish eating the <"+content+"> fishbread"<<endl;
}

int Fishbread::getCost()
{
    return cost;
}

void Fishbread::setCost(int c)
{
    cost=c;
}
```

Fishbread.cpp

```cpp
#include "Fishbread.cpp"
using namespace std;

int main()
{

Fishbread fish1(500,"red bean");
cout<<"how much? "<<fish1.getCost()<<endl;
fish1.setCost(800);
cout<<"how much? "<<fish1.getCost()<<endl;
return 0;

}
```

main.cpp

기초 복습

C++ Basics & Data Types

- 입출력
  - 출력 형태
    - Std::out << 출력대상
  - 개행
    - Std::endl;
  - 입력 형태
    - Std::cin>>'변수';

```
#include <iostream>

int main(void)
{
        int year = 2017;
        std::cout<<year<<"학년도 프로그래밍 수업"<<std::endl;
        std::cout<<"실습시간 입니다"<<std::endl;
        return 0;
}
```

C++ Basics & Data Types

- Data types
  - long double
  - double
  - float
  - unsigned long int (synonymous with unsigned long)
  - long int (synonymous with long)
  - unsigned int (synonymous with unsigned)
  - int
  - unsigned short int (synonymous with unsigned short)
  - short int (synonymous with short)
  - unsigned char
  - char
  - bool

C++ Basics & Data Types

- 문자형 char
  - char A = 'A';
- 정수형 int
  - int A = 10;
- 실수형 float, double
  - float A = 12.34;
- bool형 : true / false 가리키는 데이터형
  - bool A = 0;
  - bool A = false;
- void형 : 비어있음

C++ Basics & Data Types

- 문자형 string : 문자의 모음

String: collection of chars

  - string A = “hello”;

- 포인터 *& : 데이터의 주소를 저장하는 변수

Variable that contains address of another variable

  - int *A // 포인터 변수 생성
  - A = &B; //포인터 변수 A에 B의 주소 저장
  - *A = 100; (*A의 주소에 값 100을 저장)

- 배열 : 같은 타입의 데이터의 집합

array: collection of data of same type

  - int A[200] = {1, 2, 3}

Exercise 2 - 1

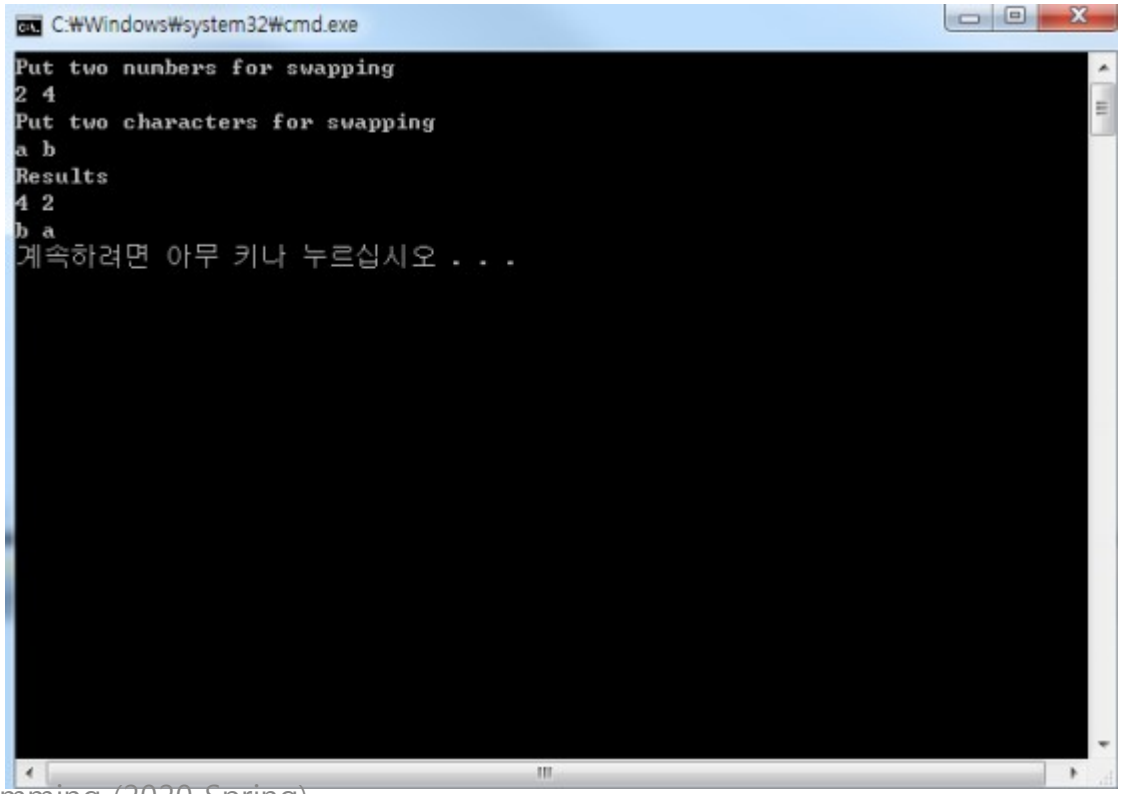
- Get two integers and two strings, print the **swapped** result.

```
#include <iostream>

using std::cout;
using std::endl;
using std::cin;

void swap(int *a, int *b);
void swap(char *a, char *b);

int main(void)
{
// FILL IN
}
void swap(int *a, int *b)
{
// FILL IN
}
void swap(char *a, char *b)
{
// FILL IN
}
```



Exercise 2 - 2

- Get a character. Swap an upper case to a lower case, Swap a lower case to an upper case. And print out swapped result.
- HINT : ASCII code : A = 65 / a= 97

```
#include <iostream>

using namespace std;

int main(void)
{
        char alphabet;
        bool out = false; // terminate when out == true

        //FILL IN

        return 0;
}
```