

## Week 3-1 : C++ Basics & Data Types

### Part1. C++ Basics

C++ : provide object orient concept in software development platform

Object-Oriented : don't think how to deal with data, but think whole data as a object

#### - Using new type "Class"

Memory size used by variable

Information owned by variable

Possible manipulation made by variable

#### - Definition of Class

New type that combines variables and associated functions

Creation a new type by defining class

Defining a structure is a one kind of creation new type, but the difference between structure and class is use of functions  
in other words, class is like the structure that contains function

#### - Keywords

Member variable : Variable in the class

Member function, Method : Function in the class, which decide object behavior

Object,Instance : A variable made by class



- Declaration of class

```
class classname
{
    Member variable;
    Member function;
};
```

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

when you declare class, memory is not allocated, but memory allocation is done when object is created.

- Definition of object

Realization of class

```
Ex)
Fishbread fish1("adzuki beans");
```

```
Fishbread *fish2=new Fishbread("custard cream");
```

**- Access the object member**

```
Ex)
fish1.content="strawberry";
fish2.wrapped();
```

```
Ex)
#include <iostream>
#define WELL_DONE 3;
class Fishbread
{
public:
    string content;
    int roasting;
};
int main()
{
    Fishbread fish1;
    fish1.content="custard cream";
    fish1.roasting=WELL_DONE;
    std::cout<<"fishbread is made of"<<fish1.content<<", and is
roasted as much as"<<fish1.roasting<<std::endl;
    return 0;
}
```

**- Access Control**

public : able to access from the class outside

protected : Inheritance relation, allow derived class to access

private : only able to access from the class inside,

if not specified, then default is public

access control to the object data for preventing unapproved access from outside (encapsulation). If you want to access to encapsulated data, then declare the set function and get function by using public identifier.

```
Ex)
class Fishbread
{
private:
int cost;
int seller;
    string content;
int roasting;
public:
    int GetCost();
    void SetCost(int cost);
};
...
fish1.SetCost(500);
cout<<"How much is it?"<<fish1.GetCost()<<endl;
fish1.SetCost(700);
cout<<"How much is it?"<<fish1.GetCost()<<endl;
```

#### - Implementation of class method

```
Ret.type class name::function name(para1,param2,..)
{
    ...
}
```

```
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;
}
int main()
{
    Fishbread fish1;
    fish1.SetCost(800);
    cout<<"How much is it"<<fish1.GetCost()<<endl;
    return 0;
}
```

**- const member method**

All member values of corresponding class cannot be changed

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

**[Exercise]**

1. using for loop, make a program which add 1 to 100.
  
2. Make a Book class, which include bookName, bookPrice variable and a price get, set function. In the main function, make a book object and print out book name, and price.