Class in C++(9th Practice)

what is c++?

○ C : high level language attribute + assembly language attribute
○ 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
○ Why make Class
  ◆To solve complex problem easily
○ 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

○ Member variable  
   ◆ Variable in the class

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

○ Object, Instance  
   ◆ A variable made by class

◆ Class  

○ Declaration of class  

class classname
{
   Member variable;
   Member function;
};

◆ Object
Ex)
class Fishbread
{
    string content;
    void Wrapped();
}; //like structure, append ‘;’ at the end of bracket

○ 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
◆ . operator: access to the member of object

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

Ex)
#include <iostream>
#define WELL_DONE 3;
class Fishbread
{
public:
```cpp
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;
}
```

- private and public identifier
- public: able to access from the class outside,

```cpp
Ex)
class Fishbread
{
    Public:
    string content;
    int roasting;
    void wrapped();
};
... 
Fishbread fish1;
fish1.content = "custard cream";
```
◆ private: only able to access from the class inside, private:

Ex)
class Fishbread{
private:
    string content;
    int roasting;
    void wrapped();
};
...
Fishbread fish1;
fish1.content="custard cream";//error! can’t access

◆ if not specified, then default is public

◆ usage: access control to the object data for preventing unapproved access from outside (encapsulation)

Ex)
class Fishbread{
    int cost; //public
    int seller; //public
    public; //public
    string content; //public
private:
    int roasting; //private
    void wrapped(); //private
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
◆ implement like defining function
◆ freely access to the private and public member data

Ret.type class name::function name(para1,parm2,..)
```cpp
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;
};

○ Constructor and Destructor
◆ Constructor
- mainly initialize the member variable of class

Ex)
Fishbread fish1;
Fishbread fish2(500,5,"adzuki bean",WELL_DONE);
-a special member function has same name as class name
- automatically called when object is made
- user cannot call it
- no return type (not even void type)
- having parameter as much as it is necessary
- function overloading is possible
- access identifier should be public

class Fishbread
{
    public:
        Fishbread(int cost);
        Fishbread(int cost,string content);
    private:
        int cost;
        int seller;
        string content;
};
```cpp
Fishbread::Fishbread(int argCost)
{
    cost=argCost;
}
Fishbread::Fishbread(int argCost,string argContent)
{
    cost=argCost;
    content=argContent;
}

◆ Destructor
- destroy object value, deallocate the assigned memory
- has a name like "~class” Ex)~Fishbread()
- user cannot call it
- no return type(not even void type)
- no parameter //difference between constructor and destructor
- cannot overload //difference between constructor and destructor
- access identifier should be public

Fishbread::~Fishbread()
{
    ...
    free(p); //object creation and free the allocated memory
    cout<<"sold out (destructor)"<<endl;
```
const member function

- when you declare ‘const’ to the member variable
  
  - all member values of corresponding class cannot be changed

```cpp
class Fishbread
{
    public:
        void SetCost() const;
    Private:
        int cost;
};

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

class declaration and making member function

- interface between class and user
- notice the variation of class data type/function
- using *.hpp file //like C, you can use *.h

definition of function

- definition of concrete function processing
  - the part to be written by programmer who want to make class
- using *.cpp file
/*Fishbread.hpp*/
#include <iostream>
class Fishbread
{
    public:
        Fishbread(int argCost,string argContent);
        ~Fishbread();
        int GetCost();
        void SetCost();
    private:
        int cost;
        string content;
};

/*Fishbread.cpp*/
#include “Fishbread.hpp”
Fishbread::Fishbread()
{
}
Fishbread::Fishbread(int argCost,string argContent)
{
    cost=argCost;
    content=argContent;
}
Fishbread::~Fishbread()
{
    cout>>“I ate the fishbread“<<endl;}
```cpp
/*main.cpp*/
#include "Fishbread.hpp"
int main()
{
    Fishbread fish1(500, "adzuki beans");
cout << "how much is it?" << fish1.GetCost() << endl;
    fish1.SetCost(800);
cout << "how much is it?" << fish1.GetCost() << endl;
    return 0;
}

[Practice]
Download project file(Lab9_Exercise.zip) from http://dcslab.snu.ac.kr/courses/cp2014s. these project uses the calculator class and it is defined in header.h. look through header.h file and define the necessary member function.

<Note> header.h main.cpp shouldn’t be modified >
<ceil means ceiling, floor means flooring, round means rounding the nearest integer>