

Lab #06

Date: 2014/04/09 – Michael Hsu

goo.gl/50ulfX

50ulfX

Recall Struct

Usually has two type members

- ✓ Data member
- ✓ Member function

brief introduction to OO

What is OO

- ✓ OO : object oriented
- ✓ represents concepts as "objects" that have data fields (attributes that describe the object) and associated procedures known as methods
- ✓ Hiding components and handling object by methods
- ✓ You don't need to know how the methods implement.
- ✓ Re-use, easy to maintain and modify.

Class

Similar to struct

But the members are private members by default (data hiding)

Class example: Student BMI

Create a Student class ...

See the file “ClassStudent.cpp”.

Class example: Visibility modifiers

```
class Student
{
    private:
        string name;
        float height;
        float weight;
    public:
        Student(); // constructor
        Student(string n, float h, float w); // constructor
        ~Student(); // destructor

        float BMI();
        void print();
};
```

Private:

1. Default
2. It can be accessed only in the class

Class example: Constructor / Destructor

```
Student::Student()  
{  
    name = "unknown";  
    height = 1.8;  
    weight = 80;  
}
```

```
Student::Student(string n, float h, float w)  
{  
    name = n;  
    height = h;  
    weight = w;  
}  
Student::~~Student()  
{  
}
```

Constructor:

1. Initialize the object with given arguments
2. Name is the same as the class
3. You can overload the constructor to do several initialization ways

Class example: Instance function

```
float Student::BMI()  
{  
    return weight / (height * height);  
}  
void Student::print()  
{  
    cout << "Hello, " << name;  
    cout << "Your BMI is " << BMI() << endl;  
}
```

In an instance function, we can invoke an instance function

Class example: Student main

```
int main()
{
    Student std1;
    std1.print();

    Student std2("Michael Hsu", 1.70, 80);
    std2.print();

    cout << "Average BMI = ";
    cout << (std1.BMI() + std2.BMI())/2 << "." << endl;

    return 0;
}
```

```
Hello, unknown. Your BMI is 24.6914.
Hello, Michael Hsu. Your BMI is 27.6817.
Average BMI = 26.1865.
```

Class example: Student main

```
int main()
{
    Student std1;
    std1.print();

    Student std2("Michael Hsu", 1.70, 80);
    std2.print();

    cout << "Average BMI = ";
    cout << (std1.BMI() + std2.BMI())/2 << "." << endl;

    return 0;
}
```

When did you call constructor?

```
Hello, unknown. Your BMI is 24.6914.
Hello, Michael Hsu. Your BMI is 27.6817.
Average BMI = 26.1865.
```

Class example: Student main

```
int main()
{
    Student std1;
    std1.print();

    Student std2("Michael Hsu", 1.70, 80);
    std2.print();

    cout << "Average BMI = ";
    cout << (std1.BMI() + std2.BMI())/2 << "." << endl;

    return 0;
}
```

invoked automatically when
the object is created.

When did you call constructor?

```
Hello, unknown. Your BMI is 24.6914.
Hello, Michael Hsu. Your BMI is 27.6817.
Average BMI = 26.1865.
```

Class example: Setter

```
void Student::setHeight(float h)
{
    height = h;
}
void Student::setWeight(float w)
{
    weight = w;
}
```

```
Student std1;
std1.setWeight(100);
std1.print();
```

Creating class in separate files

```
+--- Project Folder
```

```
    +--- Student.dev // for Dev-C++
```

```
    +--- Student.h   // Class header file
```

```
    +--- Student.cpp // Class implement file
```

```
    +--- main.cpp    // main function file
```

Lab Work: LAB14-06 Class implement(1/2)

Description

Implement a Class of role in RPG games, and that the two people attack each other for many rounds. The first people attack first. Find who will survive at the end of game.

Input

The input contains one line of eight values. In each line, there will be two roles join the game. Each role with three attributes: Health, damage and armor. The first value of each people is the name of the role. Values are separated by white spaces.

For example, a line of testing data contains

name₁ Health₁ damage₁ armor₁ name₂ Health₂ damage₂ armor₂

Output

Print the winner's name in one line of each PKs.

Sample input

Ezreal 5 3 1 Caitlyn 5 2 1

Nami 10 10 5 Nunu 20 10 9

Sample output

Ezreal

Nunu

Class: Role

```
class Role
{
private:
    string name;
    int Health;
    int damage;
    int armor;
public:
    Role();
    Role(string n, int hp, int d, int a);
    ~Role();
    bool isSurvive();
    int attack();
    void defense(int damage);
    void print();
    string getName();
};
```

You have to implement all the functions.

Q&A