

# Basics: From C to C++

**Computer Programming for Engineers (DSAF003-42)**

Fall, 2021

## **Practice 5 : Class**

**Instructor:**

Youngjoong Ko (nlp.skku.edu)

# User Define Structure Example

- very similar to the C-style structure
  - doesn't require typedef for define type

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  struct User
7  {   string ID;
8      string password;};
9  int main()
10 {
11     User user = { "cg", "skku" };
12     cout << "ID: " << user.ID << endl;
13     cout << "PASS: " << user.password << endl;
14     User* pUser = &user; //pointer access
15     cout << "ID: " << pUser->ID << endl;
16     cout << "PASS: " << pUser->password << endl;
17     return 0;
18 }
```

```
ID: cg
PASS: skku
ID: cg
PASS: skku
```

# User Define Class Example

- very similar to the C-style structure
  - doesn't require typedef for define type

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  class User
7  {
8  public: string ID;
9         string password;};
10 int main()
11 {
12     User user = { "cg", "skku" };
13     cout << "ID: " << user.ID << endl;
14     cout << "PASS: " << user.password << endl;
15     User* pUser = &user; //pointer access
16     cout << "ID: " << pUser->ID << endl;
17     cout << "PASS: " << pUser->password << endl;
18     return 0;
19 }
```

```
ID: cg
PASS: skku
ID: cg
PASS: skku
```

# Exercise 1

- Define class 'Student' described below
  - 3 member variable : ID(integer), Name(String), Major(String)
- Write function that get Student class as argument and show it's contents
  - class can passed as argument like other types

output example

```
ID: 2021711919  
Name: Taehun  
Major: Artificial Intelligence
```

main function

```
int main(){  
    Student student = {2021711919,"Taehun","Artificial Intelligence"};  
    show(student);  
}
```

# Class Member Functions Example

## ■ Class can get member functions

- member functions can access in member variables
- **this** pointer calling object itself

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  class User
7  {
8  public:
9      int ID;
10     string password;
11     void welcome(); //declaration only
12 };
13 //definition here
14 void User::welcome() {
15     cout<<"Welcome, "<< ID << endl;
16     cout<<"Password: "<< this->password << endl;
17 }
18 int main()
19 {
20     User user = { 2021711919, "skku" };
21     user.welcome();
22 }
```

```
Welcome, 2021711919
Password: skku
```

# Scope Resolution Operator Example

- Used to specify "of what thing" they are members

```
1  #include <iostream>
2
3  using namespace std;
4
5  class Angle
6  {
7  public:
8      static const double PI;
9      bool isDeg;
10     double angle;
11 };
12
13 const double Angle::PI = 3.141592;
14
15 int main()
16 {
17     Angle angle = { true, 180 };
18     //access member in class definition
19     cout << "PI: " << Angle::PI << endl;
20     cout << "PI: " << angle.PI << endl;
21     //access member in class object
22     // cout << "angle: " << Angle:angle << endl;
23     cout << "angle: " << angle.angle << endl;
24     return 0;
25 }
```

```
PI: 3.14159
PI: 3.14159
angle: 180
```

# Exercise 2

## ■ Add member functions in Angle Class

- Angle Class has 3 member variable : PI, isDeg, angle
- $1 \text{ rad} = 180 / \pi (=3.141592)$
- void RtoD : if isDeg's value is false, convert ang into degree
- void DtoR : if isDeg's value is true, convert ang into radian
- void show : show ang value
  - if value is in degree, append : °
  - if value is in radian, append : rad

main code and output example

```
const double Angle::PI = 3.141592;
int main(){
    bool select; //0(false) 1(true)
    double deg;
    cout << "Enter isdeg and angle: ";
    cin >> select >> deg;
    Angle angle = {select,deg};
    angle.show();
    angle.DtoR();
    angle.show();
    angle.RtoD();
    angle.show();
    return 0;
}
```

```
Enter isdeg and angle: 1 90.0
90°
1.5708rad
90°
```

```
Enter isdeg and angle: 0 1.5708
1.5708rad
1.5708rad
90.0002°
```

# Access Control Example

- Public variables can be accessed and modified by anyone
- Private variables only can be accessed & modified by member function

```
1  #include <iostream>
2  using namespace std;
3  class Angle{
4  public:
5      double getVal();
6      void setVal(double angle, bool isDeg);
7      bool isDeg;
8  private:
9      double angle;
10 };
11 double Angle::getVal() {
12     return angle;
13 }
14 void Angle::setVal(double angle, bool isDeg)
15 {
16     this->isDeg = isDeg;
17     this->angle = angle;
18 }
19 int main(){
20     Angle angle;
21     angle.setVal(90.0, true);
22     cout << "angle: " << angle.getVal() << endl;
23     //angle.angle = 45.0;
24     return 0;
25 }
```

angle: 90



# Assignment

## ■ Use Exercise2 code, previous code and write main function

- Class Angle contains public variables(isDeg, PI), private variable(angle)
- Class Angle contains 5 member functions(RtoD, DtoR, show, getVal, setVal)

## ■ In main function

- User inputs start angle and that start angle is degree or not
- Print value with both metric(rad &  $^{\circ}$  ), first inputted metric is first
- Angle of rotation is inputted and print rotated value with both metric, degree metric is first

output examples

```
start angle: 180
is degree?(y/n): y
180°
3.14159rad
angle of rotation: 90
270°
4.71239rad
```

```
start angle: 3.141592
is degree?(y/n): n
3.14159rad
180°
angle of rotation: -90
90°
1.5708rad
```