

C++

PROGRAMMING LANGUAGE

L07-STRUCTS

Mohammad Shaker

mohammadshaker.com

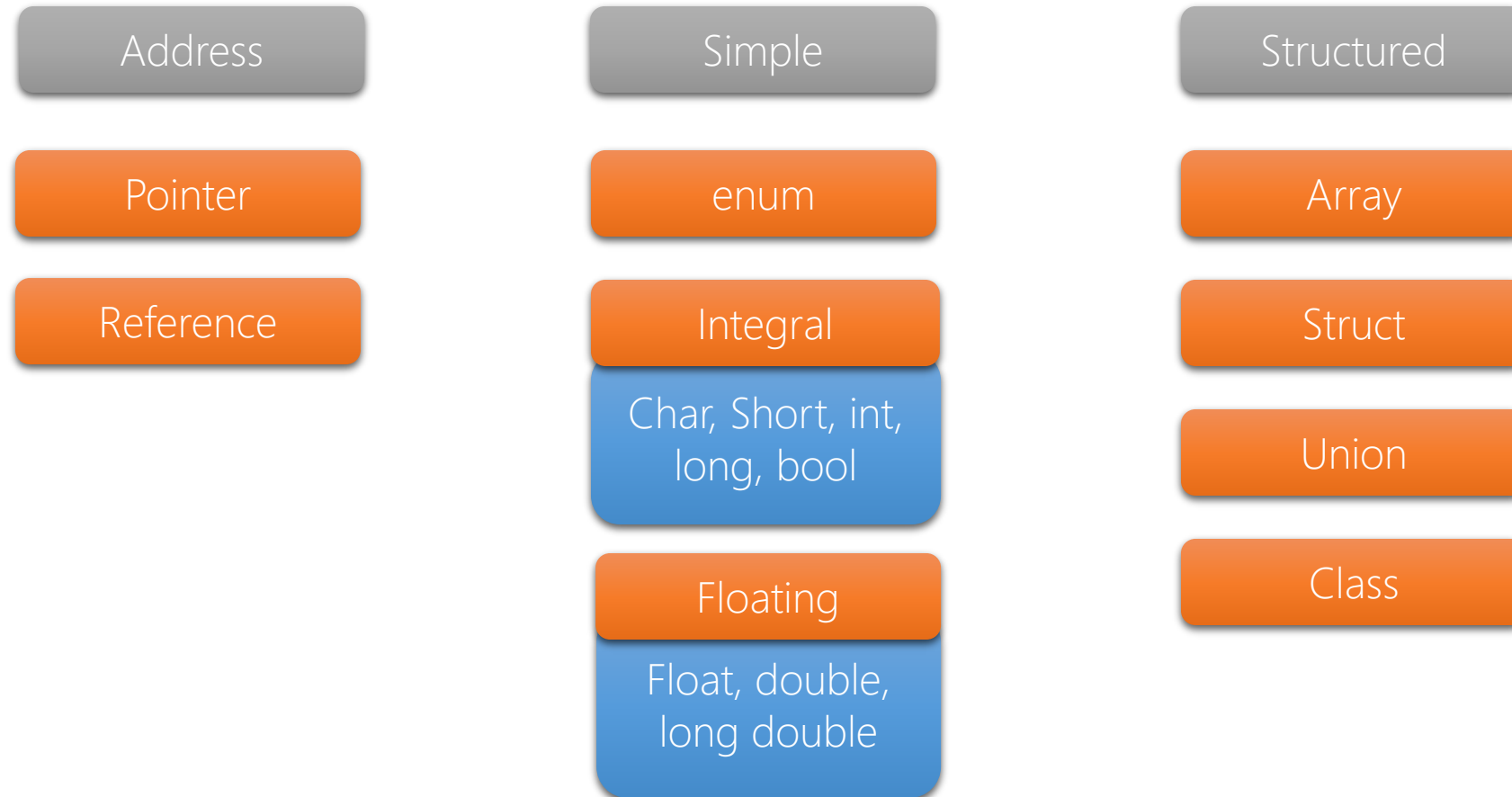
@ZGTRShaker

2010, 11, 12, 13, 14



Structs

C++ data types



Structs

- Differentiate:
 - Array:
 - Is a collection of variables of the same type
 - Struct "Structure"
 - Is a collection of variable of one or more type
- **Struct** is definition not a declaration
 - No memory allocation
 - "Memory is allocated just when we declare a variable"
 - Examples:
 - Student record, banks, players, addresses

Structs

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

void main(void)
{
    struct MyInner
    {
        int Key;
        int Salary;
    };

    struct Employee
    {
        MyInner info;
        char Name[20];
    };
}
```

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

void main(void)
{
    int i = 0;
    struct Employee
    {
        int a, b, c;
    } TempStruct [2] = {{2,4,5}, {1,6,8}};

    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

```
2
4
5
Press any key to continue
```

Structs

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

void main(void)
{
    int i = 0;
    struct Employee
    {
        int a, b, c;
    } TempStruct [2] = {{2,4}, {1,6,8}};

    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

```
2
4
0
Press any key to continue
```

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

void main(void)
{
    int i = 1;
    struct Employee
    {
        int a, b, c;
    } TempStruct [2] = {{2,4} };

    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

```
0
0
0
Press any key to continue
```

Structs

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

void main(void)
{
    int i = 1;
    struct Employee
    {
        int a, b, c;
    };
    Employee TempStruct [2] = {{2,4} };

    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

```
0
0
0
Press any key to continue
```

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

void main(void)
{
    int i = 1;
    struct Employee
    {
        int a, b, c;
    } TempStruct [2] = {{2,4} }

    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

```
Compiler error.
Missing semi colon be4 cout ";" (After TempStruct)
```

Structs

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

void main(void)
{
    int i = 1;
    struct Employee
    {
        int a, b, c;
    } TempStruct [2] = {{2,4} }

    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

Compiler error.
Missing semi colon be4 cout ";" (After TempStruct)

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

void main(void)
{
    int i = 0;
    struct
    {
        int a, b, c;
    } TempStruct [2] = {{2,8} };

    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

2
8
0
Press any key to continue

Structs

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

void main(void)
{
    int i = -1;
    struct
    {
        int a, b, c;
    } TempStruct [2] = {{2,8} };
    TempStruct[i].c = 67;
    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

Compiler error. When not writing the name of the struct we should declare variable immediately

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

void main(void)
{
    int i = -1;
    struct
    {
        int a, b, c;
    } TempStruct [2] = {{2,8} };
    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

```
2030155668
1
1698745032
Press any key to continue
```

Structs

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

void main(void)
{
    int i = 0;
    struct
    {
        int a, b, c;
    };
    TempStruct [2] = {{2,8} };

    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

```
1231378292
1
1689373384
Press any key to continue
```

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

void main(void)
{
    int i = 1;
    struct
    {
        int a, b, c;
    } TempStruct [2] = {{2,8} };
    TempStruct[i].c = 67;
    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

```
0
0
67
Press any key to continue
```

Structs

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

void main(void)
{
    int i = 1;
    struct
    {
        int a, b, c;
    } TempStruct [2] = {{2,8} };
    TempStruct.c = 67;
    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

Compiler error

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

void main(void)
{
    int i = 1;
    struct
    {
        int a, b, c;
    } TempStruct [2] = {{2,8} };
    TempStruct[i].c = 67;
    cout << TempStruct << endl;
}
```

001BF048
Press any key to continue

Structs

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

void main(void)
{
    struct student
    {
        char Name[20];
        int Num;
        int id;
    };
    student st1 = {"mee", 196, 7};
}
```

Compile and run

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

void main(void)
{
    struct student
    {
        char Name[20];
        int Num;
        int id;
    };
    student st1 = {"mee", 196, 7};
    st1 = {"zee", 111, 4};
}
```

Compiler error

Structs

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

void main(void)
{
    struct student
    {
        char Name[20];
        int Num;
        int id;
    };
    student st1 = {"mee", 196, 7};
    student st2 = {"zee", 111, 4};
    student Tempst1 = {"mee", 196, 7};

    if (st1.Num == Tempst1.Num)
    {
        cout << "Yeah!" << endl;
    }
    else
    {
        cout << "No!" << endl;
    }
}
```

Yeah!

because Structs are "value" type unlike classes which are "reference" type

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

void main(void)
{
    struct student
    {
        char Name[20];
        int Num;
        int id;
    };
    student st1 = {"mee", 196, 7};
    student st2 = {"zee", 111, 4};
    student Tempst1 = {"mee", 196, 7};

    if (st1 == Tempst1)
    {
        cout << "Yeah!" << endl;
    }
    else
    {
        cout << "No!" << endl;
    }
}
```

Compiler error

Structs

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

void main(void)
{
    struct student
    {
        char Name[20];
        int Num;
        int id;
    };
    student st1 = {"mee", 196, 7};
    student st2 = {"zee", 111, 4};
    student Tempst1 = {"mee", 196, 7};

    if (st1.Num == st2.Num)
    {
        cout << "Yeah!" << endl;
    }
    else
    {
        cout << "No!" << endl;
    }
    cout << st1.Num << endl;
    cout << st2.Num << endl;
}
```

```
Yeah!
111
111
```



structs and Pointers

Structs

```
#include <iostream>
using namespace::std;
void main(void)
{
    struct Inner
    {
        int Num;
        int id;
    };

    struct student
    {
        char Name[20];
        Inner info;
    };

    student st1 = {"mee", { 11, 3 } };
    student st2 = {"zee", 111, 4};
    student &Tempst1 = st1;
    cout << st1. Name << endl;
    cout << st1.info. Num++ << endl;
    cout << st1.info.id << endl;
    cout << ++Tempst1.info. Num << endl;
}
```

```
mee
11
3
13
```

```
#include <iostream>
using namespace::std;
void main(void)
{
    struct Inner
    {
        int Num;
        int id;
    };

    struct student
    {
        char Name[20];
        Inner info;
    };

    student st1 = {"mee", { 11, 3 } };
    student st2 = {"zee", 111, 4};
    student &Tempst1 = st1;
    cout << st1. Name << endl;
    cout << st1. Num++ << endl;
    cout << st1.info.id << endl;
    cout << ++Tempst1.info.Num << endl;
}
```

Compiler error

Structs

```
#include <iostream>
using namespace::std;
void main(void)
{
    struct Inner
    {
        int Num;
        int id;
    };

    struct student
    {
        char Name[20];
        Inner info;
    };

    student st1 = {"mee", 11, 3 };
    student st2 = {"zee", 111, 4};
    student &Tempst1 = st1;

    cout << st1. Name << endl;
    cout << st2.info. Num++ << endl;
    cout << st2. info. id << endl;
    cout << ++Tempst1. info. Num << endl;
}
```

```
mee
111
4
12
```

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

void main(void)
{
    struct Inner
    {
        int Num;
        int id;
    };

    struct student
    {
        string Name;
        Inner info;
    };

    student st1 = {"mee", 11, 3 };
    student st2 = {"zee", 111, 4};
    student &Tempst1 = st1;

    st1.Name = "foo";
    cout << st1.Name << endl;
}
```

Compiler error

Structs

```
#include <iostream>
#include <string>

using namespace::std;

void main(void)
{
    struct Inner
    {
        int Num;
        int id;
    };

    struct student
    {
        string Name;
        Inner info;
    };

    student st1 = {"mee", 11, 3 };
    student st2 = {"zee", 111, 4};
    student &Tempst1 = st1;

    st1.Name = "foo";
    cout << st1.Name << endl;
}
```

foo

```
#include <iostream>
#include <string>

using namespace::std;

void main(void)
{
    struct Inner
    {
        int Num;
        int id;
    };

    struct student
    {
        string Name;
        Inner info;
    };

    student st1 = {"mee", 11, 3 };
    student st2 = {"zee", 111, 4};
    student &Tempst1 = st1;

    st1.Name = "foo";
    cout << st1 << endl;
    cout << st1.Name << endl;
}
```

Compiler error



Struct as Function Parameter

Structs

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

struct Inner{int Num;int id;};
struct student{string Name;Inner info;};

student Print(student stu)
{
    cout << stu.Name << endl;
    return stu;
}

void main(void)
{
    student st1 = {"mee", 11, 3 };
    student st2 = {"zee", 111, 4};
    student &Tempst1 = st1;
    Print(st1);
}
```

mee

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

```
student Print(student stu)
{
    cout << stu.Name << endl;
    return stu;
}
```

```
void main(void)
{
    struct Inner
    {int Num;int id;};
    struct student
    {string Name;Inner info;};

    student st1 = {"mee", 11, 3 };
    student st2 = {"zee", 111, 4};
    student &Tempst1 = st1;
    Print(st1);
}
```

Compiler error
Inner \ student are
in main and not
global to let the
function see them!

Structs

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

struct Inner
{
    int Num;
    int id;
};

struct student
{
    string Name;
    Inner info;
};

student Print(student stu)
{
    cout << stu.Name << endl;
    stu.Name = "GoGo";
    return stu;
}

void main(void)
{
    student st1 = {"mee", 11, 3 };
    student st2 = {"zee", 111, 4};
    student &Tempst1 = st1;
    Print(st1);
    cout << st1.Name << endl;
}
```

mee
mee

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

```
struct Inner
{
    int Num;
    int id;
};

struct student
{
    string Name;
    Inner info;
};
```

```
student Print(student &stu)
{
    cout << stu.Name << endl;
    stu.Name = "GoGo";
    return stu;
}
```

```
void main(void)
{
    student st1 = {"mee", 11, 3 };
    student st2 = {"zee", 111, 4};
    student &Tempst1 = st1;
    Print(st1);
    cout << st1.Name << endl;
}
```

mee
GoGo

Structs

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

struct Inner
{int Num;int id;};

struct student
{string Name;Inner info; };

student Go()
{
    student stu;
    cout << stu.Name << endl;
    stu.Name = "GoGo";
    return stu;
}

void main(void)
{
    student st1 = {"mee", 11, 3 };
    student st2 = {"zee", 111, 4};
    st1=st2;
    cout << st1.Name << endl;
    student &Tempst1 = st1;
    st1=Go();
    cout << st1.Name << endl;
}
```

zee

GoGo



Pointers and Structs

Pointers and Structs

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

Compile and run

```
struct Inner
{int Num;int id;};

struct student
{string Name;          Inner info;};

void main(void)
{
    student st1 = {"mee", 196, 7};
    student *ptr = &st1;
    cout << ptr << endl;
}
```

```
struct Inner
{int Num;int id;};
```

0025F27C
mee
7

```
struct student
{string Name;          Inner info;};

void main(void)
{
    student st1 = {"mee", 196, 7};
    student *ptr = &st1;
    cout << ptr << endl;
    cout << ptr->Name << endl;
    cout << ptr->info.id << endl;
}
```

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

Compiler error

```
struct Inner
{int Num;int id;};

struct student
{string Name;          Inner info;};

void main(void)
{
    student st1 = {"mee", 196, 7};
    student *ptr = &st1;
    cout << ptr << endl;
    cout << *ptr << endl;
}
```

```
struct Inner
{int Num;int id;};
```

Compiler error, -> before info

```
struct student
{string Name;          Inner info;};

void main(void)
{
    student st1 = {"mee", 196, 7};
    student *ptr = &st1;
    cout << ptr << endl;
    cout << ptr->Name << endl;
    cout << ptr->info->id << endl;
}
```


Pointers and Structs

```
struct Inner
{int Num;int id;;}
```

Compiler error

```
struct student
{string Name;          Inner info;;}
```

```
void main(void)
{
    student st1 = {"mee", 196, 7};
    student *ptr = &st1;
    cout << ptr << endl;
    cout << *(ptr->Name) << endl;
    cout << *(ptr->info.id) << endl;
}
```

```
struct Inner
{int Num;int id;;}
```

Compiler error

```
struct student
{string Name;          Inner info;;}
```

```
void main(void)
{
    student st1 = {"mee", 196, 7};
    student *ptr = &st1;
    cout << (*ptr)->Name << endl;
    cout << (*ptr)->info.id << endl;
}
```

```
struct Inner
{int Num;int id;;}
```

mee

7

```
struct student
{string Name;          Inner info;;}
```

```
void main(void)
{
    student st1 = {"mee", 196, 7};
    student *ptr = &st1;
    cout << (*ptr).Name << endl;
    cout << (*ptr).info.id << endl;
}
```

Pointers and Structs

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

struct Inner
{int Num; int Arr[3];};

struct student
{string Name;          Inner info;};

void Count(student stu)
{
    int total = 0;
    for (int i = 0; i<3; i++)
    {
        total += stu.info.Arr[i];
    }
    cout << total << endl;
    stu.info.Arr[0]=3;
}

void main(void)
{
    student st1 = {"mee", 196, 1,3,3};
    Count(st1);
    Count(st1);
}
```

7
7

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

struct Inner
{int Num; int Arr[3];};

struct student
{string Name;          Inner info;};

void Count(student *stu)
{
    int total = 0;
    for (int i = 0; i<3; i++)
    {
        total += stu->info.Arr[i];
    }
    cout << total << endl;
    stu->info.Arr[0]=3;
}

void main(void)
{
    student st = {"mee", 196, 1,3,3};
    student *st1 = &st;
    Count(st1);
    Count(st1);
}
```

7
9

Pointers and Structs

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

struct Inner
{int Num; int Arr[3];};

struct student
{string Name;          Inner info;};

void Count(student *&stu)
{
    int total = 0;
    for (int i = 0; i<3; i++)
    {
        total += stu->info.Arr[i];
    }
    cout << total << endl;
    stu->info.Arr[0]=3;
}

void main(void)
{
    student st = {"mee", 196, 1,3,3};
    student *st1 = &st;
    Count(st1);
    Count(st1);
}
```

7
9

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

struct Inner
{int Num; int Arr[3];};

struct student
{string Name;          Inner info;};

void Count(student &stu)
{
    int total = 0;
    for (int i = 0; i<3; i++)
    {
        total += stu.info.Arr[i];
    }
    cout << total << endl;
    stu.info.Arr[0]=3;
}

void main(void)
{
    student st1 = {"mee", 196, 1,3,3};
    Count(st1);
    Count(st1);
}
```

7
9



Structs Members and Functions

Structs Members and Functions

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

mee

Function print is implicitly inline

```
struct student
{
    string Name;
    void print()
    {
        cout << Name << endl;
    }
};

void main(void)
{
    student st1 = {"mee" };
    st1.print();
}
```

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

Compiler error

```
struct student
{
    string Name;
    void print()
    {
        cout << student.Name << endl;
    }
};

void main(void)
{
    student st1 = {"mee" };
    st1.print();
}
```

Structs Members and Functions

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

struct student
{
    string Name;
    void print()
    {
        cout << Name << endl;
    }
    void ChangeName(string str)
    {
        Name = str;
    }
};

void main(void)
{
    string NewName= "zee";
    student st1 = {"mee" };
    st1.print();
    st1.ChangeName(NewName);
    st1.print();
}
```

mee
zee

Structs Members and Functions

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

mee
foo

```
struct student
{
    string Name;
    void student::print();
    void ChangeName(string str);
};
```

Now the functions are not implicitly
inline

```
void student::print()
{
    cout << Name << endl;
}
void student::ChangeName(string str)
{
    Name = str;
}
```

```
void main(void)
{
    string NewName= "zee";
    student st1 = {"mee" };
    st1.print();
    st1.ChangeName("foo");
    st1.print();
}
```

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

mee
foomee

```
struct student
{
    string Name;
    void student::print();
    void ChangeName(string str);
};
```

Now the functions are not implicitly
inline

```
void student::print()
{
    cout << Name << endl;
}
void student::ChangeName(string str)
{
    Name = str+Name;
}
```

```
void main(void)
{
    string NewName= "zee";
    student st1 = {"mee" };
    st1.print();
    st1.ChangeName("foo");
    st1.print();
}
```

Structs Members and Functions

- **private** data members
 - Private access
 - Can be used by a few categories of functions
- **public** data members
 - Public access
 - Can be used by all code

Structs Members and Functions

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

```
zee
Your ID = 5
```

```
struct student
{
    private:
        int TempID;

    public:
        int ID;
        string Name;
        void ChangeID(int n);
        void print();
        void ChangeName(string str);
};
```

```
void student::print()
{
    cout << Name << endl; }
void student::ChangeName(string str)
{
    Name = str+Name;}
void student::ChangeID(int n)
{
    TempID = 2;
    ID = n+TempID;
    cout << "Your ID = " << ID << endl;}
```

```
void main(void)
{
    int n = 3;            string NewName= "zee";
    student st1;          st1.print();
    st1.ChangeName(NewName); st1.print();
    st1.ChangeID(n);
}
```

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

```
Compiler error
Un delcared identifier
```

```
struct student
{
    private:
        int TempID;

    public:
        int ID;
        string Name;
        void ChangeID(int n);
        void print();
        void ChangeName(string str);
};
```

```
void student::print()
{
    cout << Name << endl; }
void student::ChangeName(string str)
{
    Name = str+Name;}
void student::ChangeID(int n)
{
    TempID = 2;
    ID = n+TempID;
    cout << "Your ID = " << ID << endl;}
```

```
void main(void)
{
    int n = 3;            string NewName= "zee";
    student st1;          st1.print();
    st1.ChangeName(NewName); st1.print();
    TempID=5;
    st1.ChangeID(n);
}
```

Structs Members and Functions

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


Compiler error
Can't access private data members

```
struct student
{
    private:
        int TempID;

    public:
        int ID;
        string Name;
        void ChangeID(int n);
        void print();
        void ChangeName(string str);
};

void student::print()
{
    cout << Name << endl; }
void student::ChangeName(string str)
{
    Name = str+Name;}
void student::ChangeID(int n)
{
    TempID = 2;
    ID = n+TempID;
    cout << "Your ID = " << ID << endl;}

void main(void)
{
    int n = 3;          string NewName= "zee";
    student st1;        st1.print();
    st1.ChangeName(NewName); st1.print();
    st1.TempID=5;
    st1.ChangeID(n); }
```



Quiz

Quiz 1, 2, 3, 4

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

Compiler error Missing;

```
void main(void)
{
    int i = 1;
    struct MyFirstS
    {
        int a, b, c;
    } TempStruct [2] = {{2,4} }

    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

```
#include <iostream>
using namespace::std;
void main(void)
{
```

Compiler error

```
    string NewName = "wa3wa3"
    struct student
    {
        char Name[20];
        int Num;
        int id;
    };
    student st1 = {"mee", 196, 7};
    st1 = {NewName, 111, 4};
}
```

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

Compiler error, TempStruct has no type!

```
void main(void)
{
    int i = 0;
    struct
    {
        int a, b, c;
    };
    TempStruct [2] = {2,8,10};

    cout << TempStruct[i].a << endl;
    cout << TempStruct[i].b << endl;
    cout << TempStruct[i].c << endl;
}
```

```
struct Inner
{int Num;int id;;};
```

Compiler error, -> before info

```
struct student
{string Name;        Inner info;;};

void main(void)
{
    student st1 = {"mee", 196, 7};
    student *ptr = &st1;
    cout << ptr << endl;
    cout << ptr->Name << endl;
    cout << ptr->info->id << endl;
}
```

Quiz 5, 6

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

struct Inner
{int Num; int Arr[3];};

struct student
{string Name;          Inner info;};

void Count(student &stu)
{
    int total = 0;
    for (int i = 0; i<3; i++)
    {
        total += 2*stu.info.Arr[i];
    }
    cout << total << endl;
    stu.info.Arr[1]=-3;
}

void main(void)
{
    student st1 = {"mee", 196, 1,3,3};
    Count(st1);
    Count(st1);
}
```

14
2

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

struct student
{
    public:
        int ID;      string Name;
        void ChangeID(int n);
        void print();
        void ChangeName(string str);
};

void student::print()
{
    cout << Name << endl; }
void student::ChangeName(string str)
{
    Name = str+Name;}
void student::ChangeID(int n)
{
    static int TempID = 4;
    TempID += 2;
    ID = n+TempID;
    cout << "Your ID = " << ID << endl;}

void main(void)
{
    int n = 3;          string NewName= "zee";
    student st1;        st1.print();
    st1.ChangeName(NewName);    st1.print();
    st1.ChangeID(n);
    n = 5;
    st1.ChangeName(NewName);    st1.print();
    st1.ChangeID(n); }
```

zee
Your ID = 9
zeezee
Your ID = 13