

SDF II(15B11CI211)

EVEN Semester 2021



2nd Semester , First Year

Jaypee Institute Of Information Technology (JIIT), Noida

Lecture 4 – Classes and Objects in C++

- ✓ A **structure** is a user-defined data type available in C that allows to combining data items of different kinds. Structures are used to represent a record.

- ✓ **Struct address**

```
{    int houseno;  
    Char area[20]  
  
    Char city[20];  
    Char state[20];  
  
};  
  
address a1;
```

- A class is a way to bind data describing an entity and its associated functions together.

```
class Account
```

```
{  int account no;
    char type;
    float balance;
    float deposit (float amount)
    {    balance+= amount;
        return amount;
    }
    float withdraw (float amount)
    {    balance-= amount;
        return amount;
    } obj1, obj2
```

Comparative analysis of Structure and Class

- The main difference between structures and classes is that by default, all member of the structure are public but in class all member are by default are private.
- A structure is considered as the value type whereas, a class is a reference type.
- No structure member can have a null value. Conversely, the variables of a class can have null values.
- In order to initialize the member of a class, the constructors and destructors are used. but in the structure can initialize its members automatically.

Declaration of class

- Data member
- Member functions
- Program access
- Class tag-name
- Class definition
- Class method definition

Definition of class

```
class class-name
```

```
{
```

```
    private:
```

```
        [variable declaration]
```

```
        [function declaration]
```

```
    protected:
```

```
        [variable declaration]
```

```
        [function declaration]
```

```
    public:
```

```
        [variable declaration]
```

```
        [function declaration]
```

```
}
```

Class Method Definition

- Outside the class
- Inside the class.
- To declare method outside the class we used scope resolution operator(::)

Eg **return-type class-name::function name**

float Account:: loan()

Outside the class

```
class Account
```

```
    {   int account no;  
        char type;  
        float balance;  
    };
```

```
    float Account :: deposit (float amount)
```

```
    {   balance+= amount;  
        return amount;  
    }
```

```
    float Account :: withdraw (float amount)
```

```
    {   balance-= amount;  
        return amount;  
    }
```

Inside the class

```
class Account
```

```
{   int account no;  
    char type;  
    float balance;  
    float deposit (float amount)  
    {   balance+= amount;  
        return amount;  
    }  
    float withdraw (float amount)  
    {   balance-= amount;  
        return amount;  
    } obj1, obj2
```

Access Specifies of Class

- **The Access modifiers** of C++ allows us to determine which class members are accessible to other classes and functions, and which are not.
- There are 3 types of access modifiers available in C++:
- **Public**
- **Private**
- **Protected**

Public

- The data members and member functions declared as public can be accessed by other classes and functions.
- The public members of a class can be accessed from anywhere in the program using the direct member access operator (.) with the object of that class.

```
#include <iostream>
#include<conio.h>
using namespace std;
class Test {
public:
    int num1=10;
    int num2=20;
    void show()
    {
        cout<<"num1"<<num1<<endl;
        cout<<"num2"<<num2<<endl;
    }
}
```

```
void sum()
{
    int total;
    total=num1+num2;
    cout<<"inside sum";
    cout<<total<<endl;
} };

int main() {
    Test obj1,obj2;
    obj1.show();
    obj2.num1=5;
    obj2.num2=6
    obj2.sum();
    return 0;
}
```

Output
num1 = 10
Num2=20
Total=11

Private

The class members declared as *private* can be accessed only by the member functions inside the class.

They are not allowed to be accessed directly by any object or function outside the class


```
#include <iostream>
#include<conio.h>
using namespace std;
class Test {
private:
    int num1=10;
    int num2=20;
    void show()
    {
        cout<<"num1"<<num1<<endl;
        cout<<"num2"<<num2<<endl;
    }
}
```

```
public:
void sum()
{
    int total;
    total=num1+num2;
    cout<<"inside sum";
    cout<<total<<endl;
} };
int main() {
    Test obj1,obj2;
    obj1.show();
    obj2.num1=5;
    obj2.num2=6
    obj2.sum();
    return 0;
}
```

Output
 error: 'int Test::num1'
 is private within this
 context obj2.num1=4;

Protected

- Protected access modifier is similar to private access modifier means it can't be accessed outside of it's class unless with the help of friend class.
- This can be implemented through inheritance.(discuss it later)

Nested Classes

- A nested class is a class which is declared in another class.
- The nested class is also a member variable of the enclosing class and has the same access rights as the other members.

```
class A {  
    public:  
    class B {  
        private:  
        int num;  
        public:  
        void getdata(int n) {  
            num = n;  
        }  
        void putdata() {  
            cout<<"The number is "<<num;  
        }  
    };  
};  
int main() {  
    cout<<"Nested classes in C++"<< endl;  
    A :: B obj;  
    obj.getdata(9);  
    obj.putdata();  
    return 0;  
}
```

Internal Representation of Objects

- Object are the instance of a class.
- It does not created when we declare class.
- We create an object for class, using a class as a type specifier.

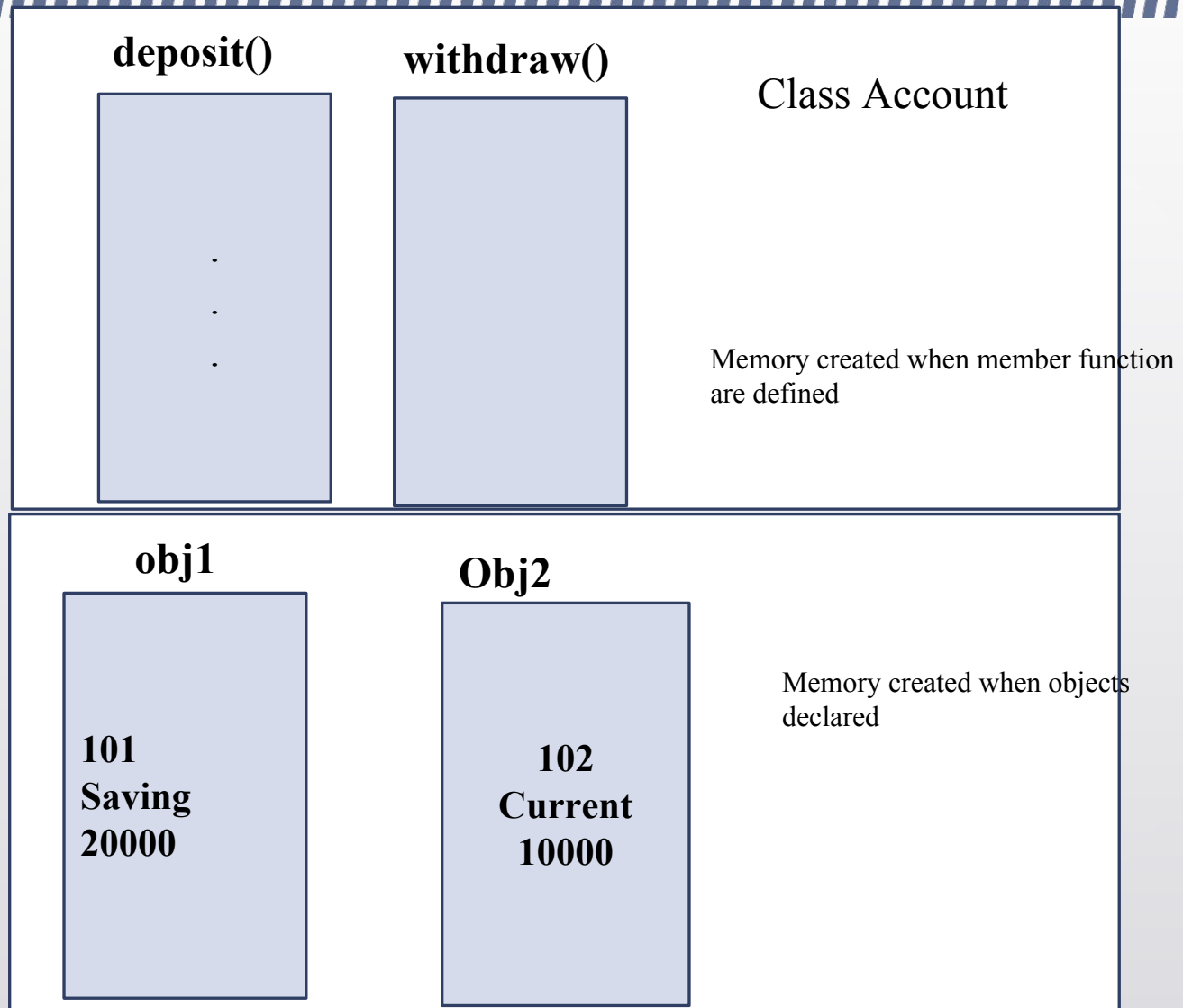
For eg

Account a1,a2;

Account is my class name and it has two objects a1 and a2.

Memory representation of the objects

- Each objects allocated memory for the its data members.
- Members functions are used by the all the object.
- Member functions are created and store when the class is defined and only once.
- All objects of the class hold different memory to store there data members as data member can have different values.



References

- <https://www.geeksforgeeks.org/access-modifiers-in-c/>
- <https://www.techprevue.com/sumita-arora-c-class-12-pdf-solutions/>
- S. Arora, Computer science in C++. 2002