Report

Aishani Anavkar

Roll no: 1806

Semester 2

Subject: C++

Topic: Objects and Classes

FYBsc. Data Science

Simple Classes

A Class is a user defined data-type which has data members and member functions.

It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class.

A C++ class is like a blueprint for an object.

Classes are the standard unit of programming

Member Functions in Class

There are 2 ways to define a member function:

- Inside class definition
- Outside class definition

To define a member function outside the class definition we have to use the scope resolution :: operator along with class name and function name.

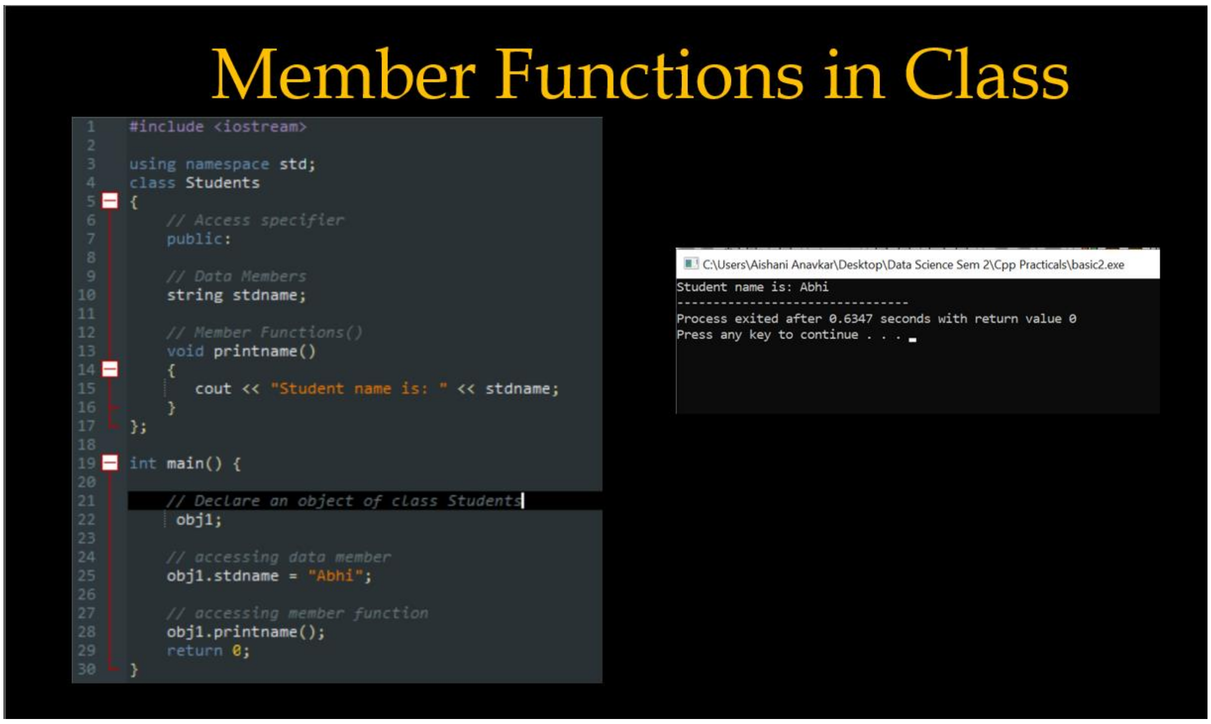
Note that all the member functions defined inside the class definition are by default inline, but you can also make any non-class function inline by using keyword inline with them.

CODE 1: Member Functions in Class

```
Member Functions in Class

#include <iostream>


using namespace std;

class Students

{

  // Access specifier

  public:
  // Data Members
  string stdname;

  // Member Functions()
  void printname()
  {
    cout << "Student name is: " << stdname;
  }
};

int main() {

  // Declare an object of class Students
   obj1;

  // accessing data member
  obj1.stdname = "Abhi";

  // accessing member function
  obj1.printname();
  return 0;
}
```

OUTPUT:

```
Student name is: Abhi
```



Member Functions in Class

```cpp
#include <iostream>

using namespace std;
class Students
{
    // Access specifier
    public:

    // Data Members
    string stdname;

    // Member Functions()
    void printname()
    {
        cout << "Student name is: " << stdname;
    }
};

int main() {

    // Declare an object of class Students
     obj1;

    // accessing data member
    obj1.stdname = "Abhi";

    // accessing member function
    obj1.printname();
    return 0;
}
```

C:\Users\Aishani Anavkar\Desktop\Data Science Sem 2\Cpp Practicals\basic2.exe

```
Student name is: Abhi
--------------------------------
Process exited after 0.6347 seconds with return value 0
Press any key to continue . . .
```

Syntax:

class class_name {

public:

constructor and destructor

member functions

private:

data members

Passing an Object as Argument

To pass an object as an argument we write the object name as the argument while calling the function the same way we do it for other variables..

Syntax: function_name(object_name);

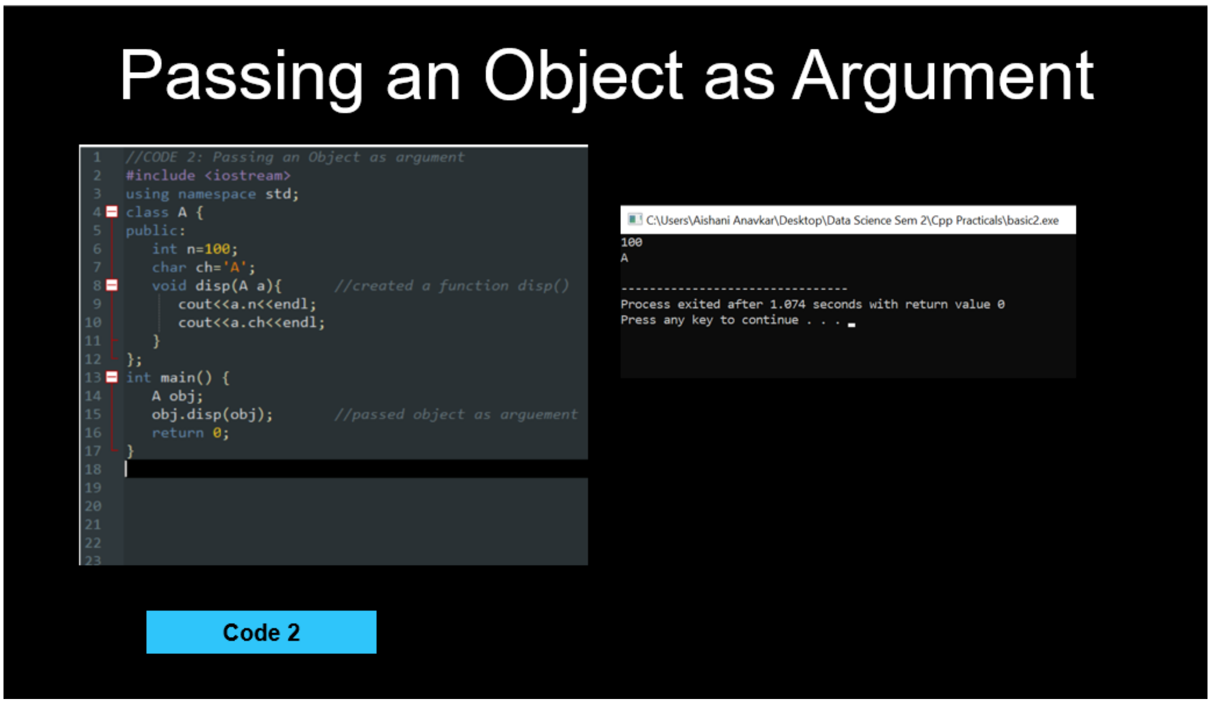
The data members of the calling object can be directly accessed inside the function without using the object name and the dot operator

CODE 2: Passing an Object as argument

```
#include <iostream>

using namespace std;

class A {

public:

  int n=100;

  char ch='A';

  void disp(A a){              //created a function disp()

    cout<<a.n<<endl;

    cout<<a.ch<<endl;

  }

};

int main() {

  A obj;

  obj.disp(obj);               //passed object as arguement

  return 0;

}
```

OUTPUT:

```
100

A
```



Passing an Object as Argument

```cpp
//CODE 2: Passing an Object as argument
#include <iostream>
using namespace std;
class A {
public:
   int n=100;
   char ch='A';
   void disp(A a){      //created a function disp()
      cout<<a.n<<endl;
      cout<<a.ch<<endl;
   }
};
int main() {
   A obj;
   obj.disp(obj);       //passed object as arguement
   return 0;
}
```

C:\Users\Aishani Anavkar\Desktop\Data Science Sem 2\Cpp Practicals\basic2.exe

```
100
A

--------------------------------
Process exited after 1.074 seconds with return value 0
Press any key to continue . . .
```

Friend Classes

Friendship is not inherited

Like friend class, a friend function can be given a special grant to access private and protected members.

The concept of friends is not there in Java.

Friendship is not mutual. If class A is a friend of B, then B doesn't become a friend of A automatically.

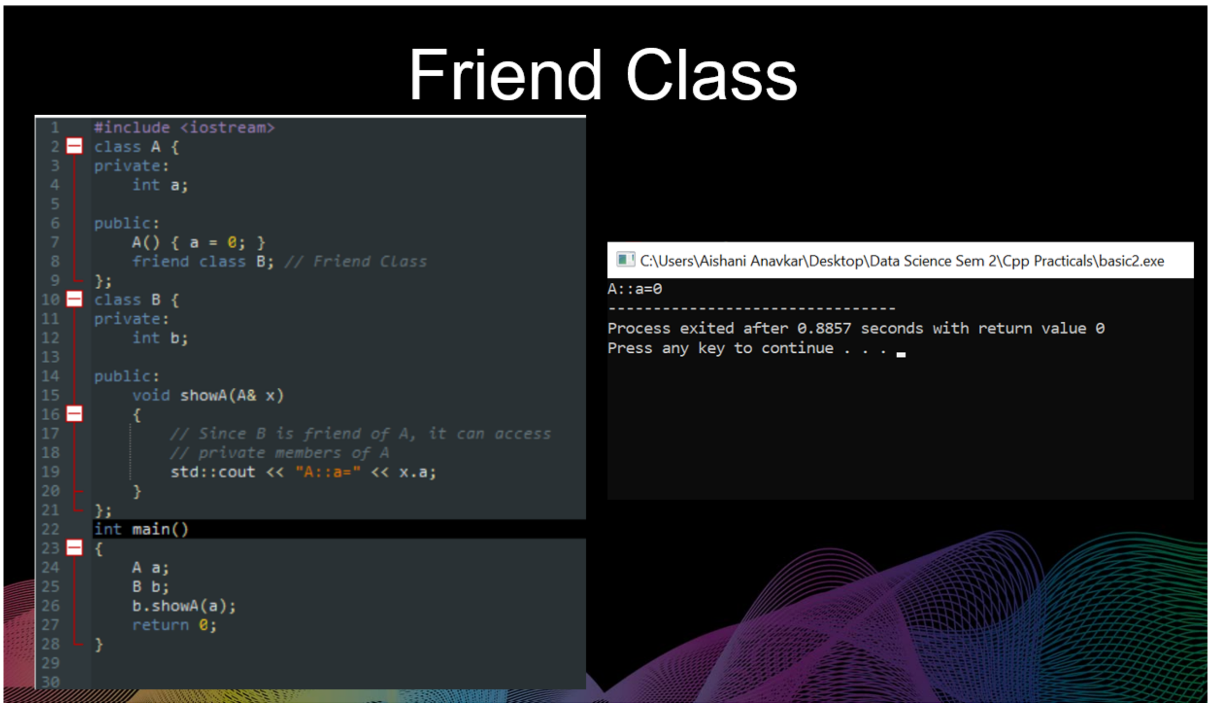
Class A friend class can access private and protected members of other class in which it is declared as friend.

CODE 3: Friend Classes

```
#include <iostream>

class A {

private:

   int a;


public:

   A() { a = 0; }

   friend class B; // Friend Class

};


class B {

private:

   int b;


public:

   void showA(A& x)

   {
        // Since B is friend of A, it can access
        // private members of A
        std::cout << "A::a=" << x.a;
    }
};

int main()
{
    A a;
    B b;
    b.showA(a);
    return 0;
}
```

OUTPUT:

```
A::a=0
```



Friend Class

```
#include <iostream>
class A {
private:
    int a;

public:
    A() { a = 0; }
    friend class B; // Friend Class
};
class B {
private:
    int b;

public:
    void showA(A& x)
    {
        // Since B is friend of A, it can access
        // private members of A
        std::cout << "A::a=" << x.a;
    }
};
int main()
{
    A a;
    B b;
    b.showA(a);
    return 0;
}
```

C:\Users\Aishani Anavkar\Desktop\Data Science Sem 2\Cpp Practicals\basic2.exe

```
A::a=0
--------------------------------
Process exited after 0.8857 seconds with return value 0
Press any key to continue . . .
```