Contents

- Classes in C++
- OOP (Object Oriented Program)
- Methods of Classes
- Examples with classes

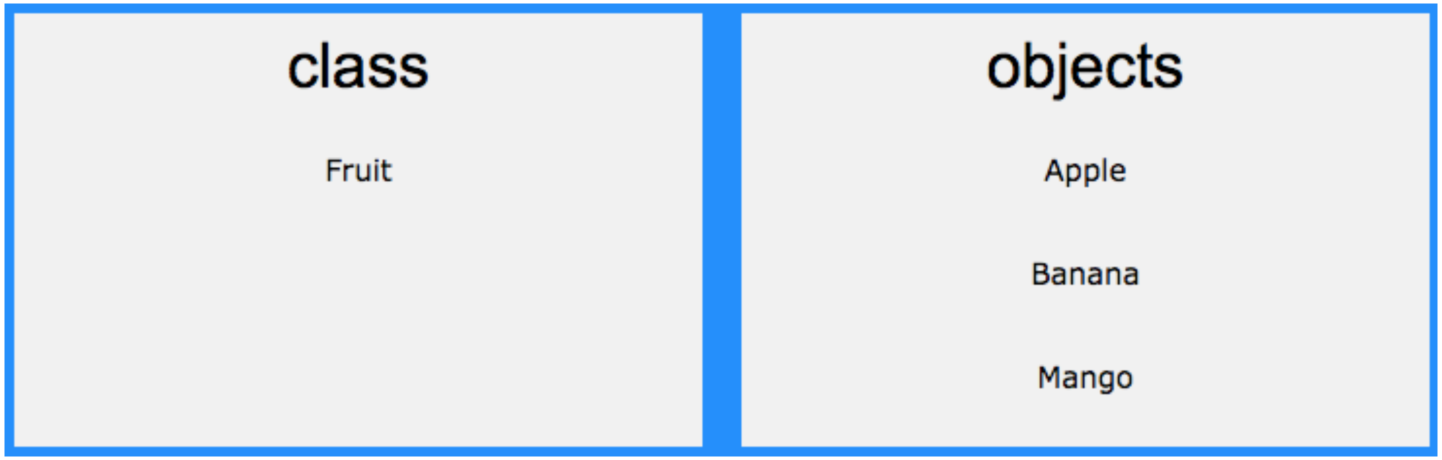
C++ Classes

What is OOP (Object-Oriented Programming)?

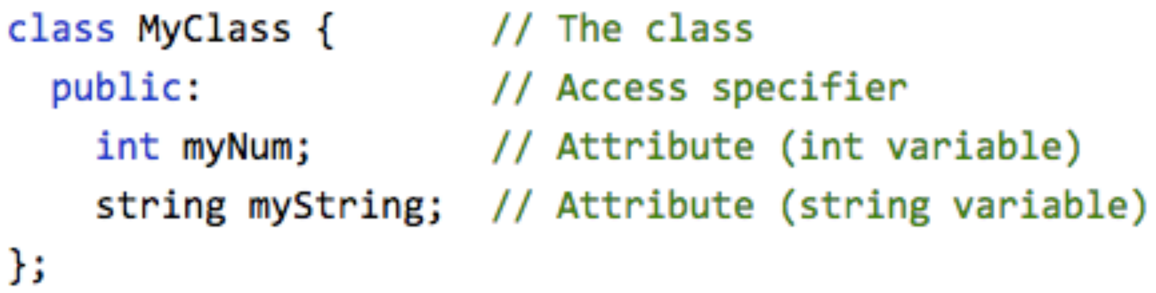
The prime purpose of C++ programming was to add object orientation to the C programming language, which is in itself one of the most powerful programming languages.

The core of the pure object-oriented programming is to create an object, in code, that has certain properties and methods.

Class of C++



Constitute of Class



```
class MyClass {        // The class
  public:              // Access specifier
    int myNum;         // Attribute (int variable)
    string myString;   // Attribute (string variable)
};
```

Ex-1: MyClass

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

class MyClass {       // The class
  public:             // Access specifier
    int myNum;        // Attribute (int variable)
    string myString;  // Attribute (string variable)
};

int main() {
  MyClass myObj;  // Create an object of MyClass
  // Access attributes and set values
  myObj.myNum = 11;
  myObj.myString = ”Last Lecture in May”;
  // Print values
  cout << myObj.myNum << "\n";
  cout << myObj.myString;
  return 0;
}
```

```
11
Last Lecture in May
```

Ex-2: Multiple objects in class

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

class Car {
  public:
    string brand;
    string model;
    int year;
};

int main() {
  Car carObj1;
  carObj1.brand = "BMW";
  carObj1.model = "X5";
  carObj1.year = 1999;

  Car carObj2;
  carObj2.brand = "Ford";
  carObj2.model = "Mustang";
  carObj2.year = 1969;

  cout << carObj1.brand << " " << carObj1.model << " " << carObj1.year << "\n";
  cout << carObj2.brand << " " << carObj2.model << " " << carObj2.year << "\n";
  return 0;
}
```

```
BMW X5 1999
Ford Mustang 1969
```

Methods of Class

There are 2 ways:

In-class definition
Out of class definition

In the example below, we define a function in the class and call it "myMethod".

Ex-3: Class Methods

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

class MyClass {         // The class
  public:               // Access specifier
    void myMethod() {   // Method/function
      cout << ”Corona virus!";
    }
};

int main() {
  MyClass myObj;     // Create an object of
MyClass
  myObj.myMethod();  // Call the method
  return 0;
}
```

```
Corona virus!
```

Ex-4: Adding parameters in the class

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

class Car {
  public:
    int speed(int maxSpeed);
};

int Car::speed(int maxSpeed) {
  return maxSpeed;
}

int main() {
  Car myObj; // Create an object of Car
  cout << myObj.speed(220); // Call the method with an argument
  return 0;
}
```

```
220
```

EX-5: To use method in class

```
/* C++ program to create class methods*/

#include <iostream>
using namespace std;

// class definition
// "Sample" is a class
class Sample {
public: // Access specifier
    // method definition 1
    void printText1()
    {
        cout << "IncludeHelp.com\n";
    }
```

Ex-6: Skipping some array elements

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

int main()
{
        int arr[10] = {1,2,3,4,5,6,7,8,9,10};

        for(int i=0; i<10; i++)
        {
          if((i+1)%3 == 0)  //If index is every third element
                continue;  //Continue
          cout<<arr[i]<<" ";  //Print array element
        }

        return 0;
}
```

```
1 2 4 5 7 8 10
```

Ex-7: Calculating the area of the rectangle using Class

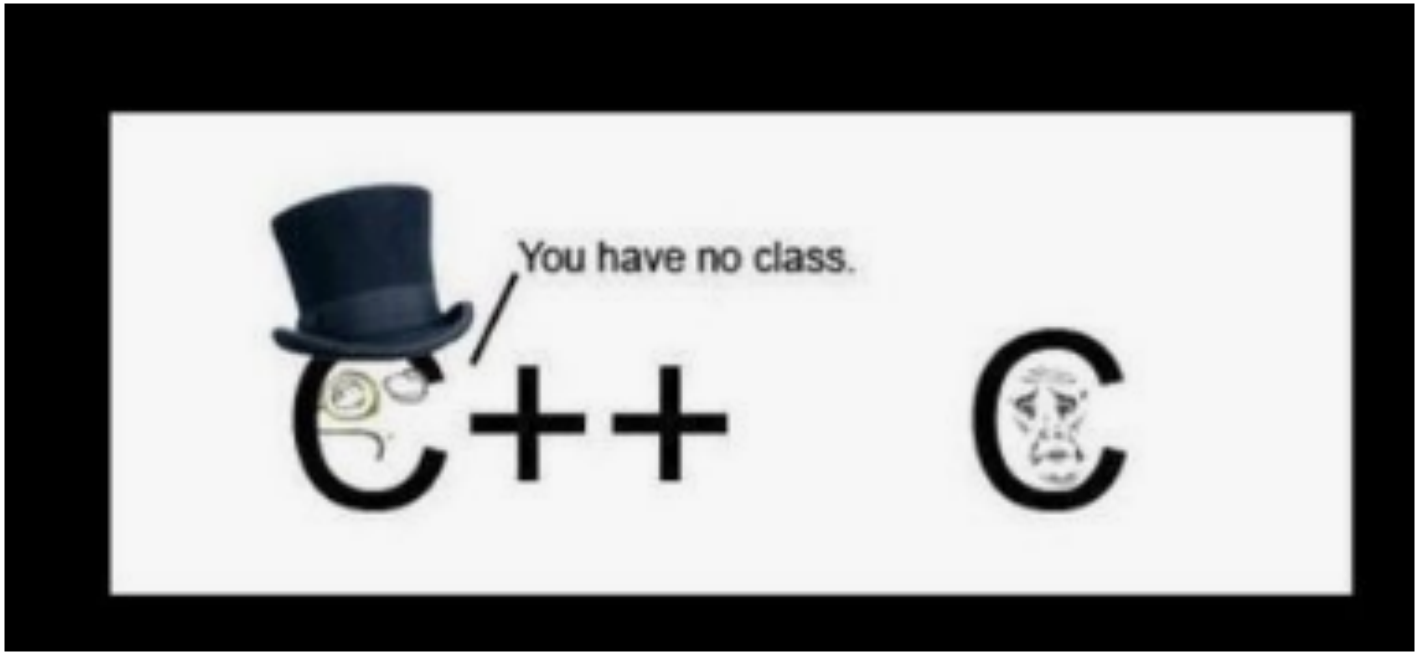
```
// classes example
#include <iostream>
using namespace std;

class Rectangle {
    int width, height;
  public:
    void set_values (int,int);
    int area() {return width*height;}
};

void Rectangle::set_vaiues (int x, int y) {
  width = x;
  height = y;
}

int main () {
  Rectangle rect;
  rect.set_values (3,4);
  cout << "area: " << rect.area();
  return 0;
}
```

```
area: 12
```



Thank you..