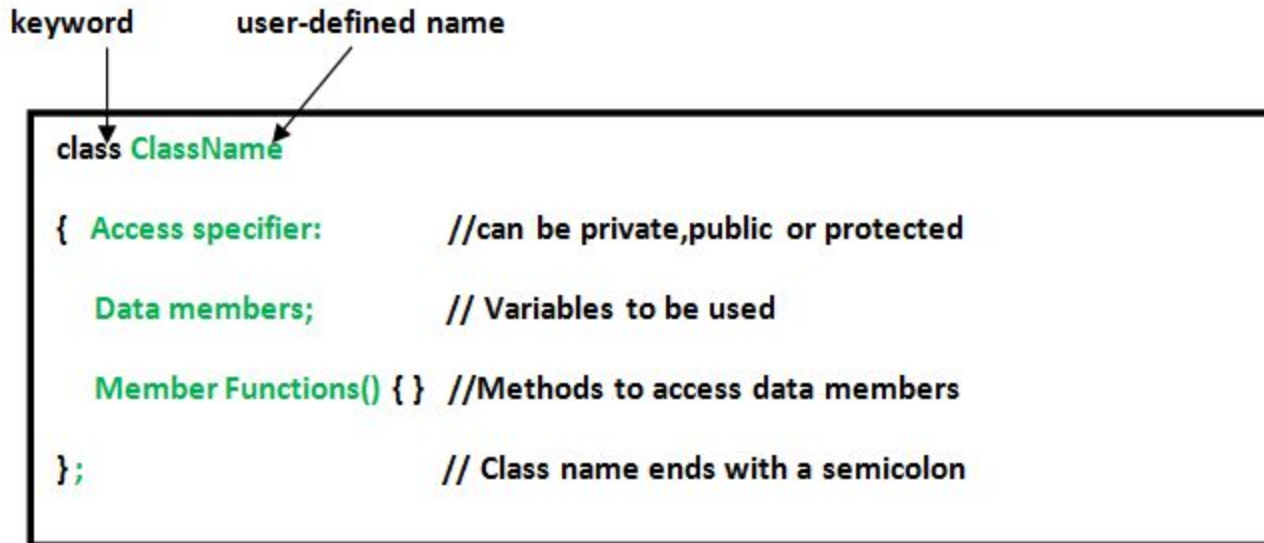


Bài 15: Class

Nguyễn Hoàng Anh

Giới thiệu

- Trong C++, từ khóa "class" được sử dụng để định nghĩa một lớp, là một cấu trúc dữ liệu tự định nghĩa có thể chứa dữ liệu và các hàm thành viên liên quan.
- Lớp là nền tảng của lập trình hướng đối tượng (OOP) trong C++



Phạm vi truy cập

- Phạm vi truy cập trong **class** là cách quy định mức độ truy cập của các thành viên (biến và phương thức) trong một lớp.
- C++ cung cấp ba phạm vi truy cập chính:
 - public
 - private
 - protected
- Mỗi phạm vi truy cập sẽ có đặc điểm riêng biệt và liên quan đến các tính chất hướng đối tượng khác nhau.

Phạm vi truy cập - public

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

class HìnhChuNhat
{
public:
    double chieuDai;    // property
    double chieuRong;   // property
};

int main()
{
    HìnhChuNhat hình1;
    hình1.chieuDai = 10.0;
    hình1.chieuRong = 5.0;
    cout << "Chieu dai: " << hình1.chieuDai<< endl;
    return 0;
}
```

Phạm vi truy cập - public

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

class HìnhChuNhat
{
public:
    double chieuDai;    // property
    double chieuRong;   // property

    // Hàm tính diện tích
    double tinhDienTich() // method
    {
        return chieuDai * chieuRong;
    }

    void display(); // method
};

void HìnhChuNhat::display()
{
    cout << " Hello " << "\n";
}

int main()
```

Phạm vi truy cập - public

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

class SinhVien
{
    public:
        int ID;          // property
        string name;      // property
        string lop;       // property

        void display()   // method
        {
            cout << "MSSV: " << ID << endl;
            cout << "TEN: " << name << endl;
            cout << "LOP: " << lop << endl;
        }
};

int main(int argc, char const *argv[])
{
    SinhVien sv; // sv được gọi là object
    sv.ID = 2010117;
    sv.name = "Anh";
```

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

class SinhVien
{
public:
    int ID;          // property
    string name;     // property
    string lop;      // property
    void display();  // method
};

void SinhVien::display()
{
    cout << "MSSV: " << ID << endl;
    cout << "TEN: " << name << endl;
    cout << "LOP: " << lop << endl;
}

int main(int argc, char const *argv[])
{
    SinhVien sv; // sv được gọi là object
    sv.ID = 2010117;
    sv.name = "Anh";
    sv.lop = "DD20TD1";
    sv.display();
}
```

File main.hpp

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

using namespace std;

class SinhVien
{
public:
    int ID;           // property
    string name;      // property
    string lop;       // property
    void display();  // method
};
```

File main.cpp

```
#include <iostream>
#include <string>
#include "main.hpp"

using namespace std;

void SinhVien::display()
{
    cout << "MSSV: " << ID << endl;
    cout << "TEN: " << name << endl;
    cout << "LOP: " << lop << endl;
}

int main(int argc, char const *argv[])
{
    SinhVien sv; // sv được gọi là object
    sv.ID = 2010117;
    sv.name = "Anh";
    sv.lop = "DD20TD1";
    sv.display();
    return 0;
}
```


Constructor

- Constructor trong C++ là một method sẽ được tự động gọi khi khởi tạo object.
- Constructor sẽ có tên trùng với tên của class.
- Thường sử dụng để khởi tạo những giá trị ban đầu.

```
class HìnhChuNhat
{
    public:
        double chieuDai;
        double chieuRong;

        HìnhChuNhat (): chieuDai (100), chieuRong (100)
        {
            chieuDai  = 10;
            chieuRong  = 9;
        }

        // Hàm tính diện tích
        double tinhDienTich ()
        {
            return chieuDai * chieuRong;
        }
}
```

Constructor

```
class HìnhChuNhat
{
    public:
        double chieuDai;
        double chieuRong;

        HìnhChuNhat();

        // Hàm tính diện tích
        double tinhDienTich()
        {
            return chieuDai * chieuRong;
        }
};

HìnhChuNhat::HìnhChuNhat()
{
    chieuDai = 10;
    chieuRong = 9;
}
```

Constructor

```
class HìnhChuNhat
{
    public:
        double chieuDai;
        double chieuRong;

        HìnhChuNhat(int dai = 5, int rong = 3);

        // Hàm tính diện tích
        double tinhDienTich()
        {
            return chieuDai * chieuRong;
        }
};

HìnhChuNhat::HìnhChuNhat(int dai, int rong)
{
    chieuDai = dai;
    chieuRong = rong;
}
```

Destructor

- Destructor trong C++ là một method sẽ được tự động gọi trước khi object được giải phóng.
- Destructor sẽ có tên trùng với tên của class và thêm ký tự ~ ở phía trước tên.
- Thường dùng để xóa dữ liệu sau khi dùng xong hoặc thu hồi vùng nhớ đã cấp phát động.

```
class HìnhChuNhat
{
    public:
        double chieuDai;
        double chieuRong;

        HìnhChuNhat ()
        {
            chieuDai = 10;
            chieuRong = 9;
        }

        ~HìnhChuNhat ()
        {
```

Static property

- Khi một property trong class được khai báo với từ khóa **static**, thì tất cả các object sẽ dùng **chung địa chỉ** của property này.

```
class HìnhChuNhat
{
    public:
        double chieuDai;
        double chieuRong;
        static int var;
};

int HìnhChuNhat::var;

int main()
{
    HìnhChuNhat hinh1;
    HìnhChuNhat hinh2;
    HìnhChuNhat hinh3;

    cout << "address of chieu dai: " << &hinh1.chieuDai << '\n';
    cout << "address of chieu dai: " << &hinh2.chieuDai << '\n';
    cout << "address of chieu dai: " << &hinh3.chieuDai << '\n';
}
```

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

class SinhVien
{
public:
    SinhVien(string name, string lop); // constructor // tự động khởi tạo
    int ID; // property
    string name; // property
    string lop; // property
    void display(); // method
};

SinhVien::SinhVien(string name, string lop)
{
    static int id = 100;
    SinhVien::ID = id;
    cout << "Create class ID: " << ID << endl;
    id++;

    SinhVien::name = name;
    SinhVien::lop = lop;
}

void SinhVien::display()
{
```

Static method

Khi một **method** trong class được khai báo với từ khóa **static**:

- Method này độc lập với bất kỳ đối tượng nào của lớp.
- Method này có thể được gọi ngay cả khi không có đối tượng nào của class tồn tại.
- Method này có thể được truy cập bằng cách sử dụng tên class thông qua toán tử `::` .
- Method này có thể truy cập các static property và các static method bên trong hoặc bên ngoài class.
- Method có phạm vi bên trong class và không thể truy cập con trỏ đối tượng hiện tại.

Static keyword

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

class HinhChuNhat
{
    public:
        double chieuDai;
        double chieuRong;
        static int count;

        HinhChuNhat(int dai = 5, int rong = 4);
        ~HinhChuNhat();

        double tinhDienTich();
        static void display();
};

HinhChuNhat::HinhChuNhat(int dai, int rong)
{
    chieuDai = dai;
    chieuRong = rong;
```