

Hàm Constructor (Hàm tạo)

Hàm tạo hay là hàm khởi tạo (constructor) là hàm được gọi khi đối tượng được khởi tạo. Trong 1 class có thể có nhiều hàm tạo tuy nhiên tham số trong các hàm tạo phải khác nhau.

```
[TÊN CLASS]([CÁC THAM SỐ]) {  
    //Các lệnh của hàm tạo  
}
```

Hàm Constructor (Hàm tạo)

```
class Line {  
    private:  
        float length;  
    public:  
        Line() {  
            this->length = 0;  
        }  
};  
...  
Line line1, line2;  
Line* line = new Line();
```

Hàm Constructor (Hàm tạo)

```
class Rectangle {  
    private:  
        float width, height;  
    public:  
        Rectangle() {  
            this->width = 10;  
            this->height = 20;  
        }  
};  
  
...  
Rectangle r1, r2;  
Rectangle* r = new Rectangle();
```

Hàm Constructor (Hàm tạo)

```
class Rectangle {  
    private:  
        float width, height;  
    public:  
        Rectangle(): width(10), height(20) {  
  
        }  
};  
  
...  
Rectangle r1, r2;  
Rectangle* r = new Rectangle();
```

Hàm Constructor (Hàm tạo)

```
class Rectangle {  
    private:  
        float width, height;  
    public:  
        Rectangle(float width, float height) {  
            this->width = width;  
            this->height = height;  
        }  
};  
  
...  
Rectangle r1(10, 20), r2(30, 40);  
Rectangle* r = new Rectangle(30, 30);
```

Hàm Constructor (Hàm tạo)

```
class Rectangle {  
    private:  
        float width, height;  
    public:  
        Rectangle(float w, float h): width(w), height(h) {  
  
        }  
};  
  
...  
Rectangle r1(10, 20), r2(30, 40);  
Rectangle* r = new Rectangle(30, 30);
```

Hàm Constructor (Hàm tạo)

```
class Rectangle {  
    private:  
        float width, height;  
    public:  
        Rectangle(): width(0), height(0) {  
        }  
        Rectangle(float w, float h): width(w), height(h) {  
        }  
};  
  
...  
Rectangle r1, r2(30, 40);  
Rectangle* r = new Rectangle();
```

Hàm Constructor (Hàm tạo)

```
class Rectangle {  
    private:  
        float width, height;  
    public:  
        Rectangle(float w): width(w), height(0) {  
        }  
        Rectangle(float w, float h): width(w), height(h) {  
        }  
};  
  
...  
Rectangle r1, r2(30, 40); // r1 lỗi  
Rectangle* r = new Rectangle(); // Lỗi
```


Hàm Destructor (Hàm huỷ)

Hàm huỷ (*destructor*) là hàm được gọi khi đối tượng bị giải phóng ra khỏi vùng nhớ. Mỗi *class* chỉ có 1 hàm huỷ duy nhất. Biến trong stack bị huỷ khi kết thúc hàm hoặc khối block. Biến trong heap bị huỷ chỉ khi sử dụng từ khoá *delete*.

```
~[TÊN CLASS]() {  
    //Các lệnh của hàm huỷ  
}
```

Hàm Destructor (Hàm huỷ)

```
class Rectangle {  
    private:  
        float *width;  
    public:  
        ~Rectangle() {  
            delete width;  
        }  
};  
  
...  
Rectangle r1, r2;  
Rectangle* r = new Rectangle();  
delete r;
```

Hàm Destructor (Hàm huỷ)

```
class Rectangle {  
    private:  
        float width;  
    public:  
        ~Rectangle() {  
        }  
};  
  
...  
Rectangle r1, r2;  
Rectangle* r = new Rectangle();  
delete r;
```