Contents

[Static and Friend trong OOP 2](#_Toc111710615)

[I. Static trong OOP 2](#_Toc111710616)

[1.1 Static variables trong OOP 2](#_Toc111710617)

[1.2. Static function trong OOP 3](#_Toc111710618)

[II. Từ khóa friend trong OOP 6](#_Toc111710619)

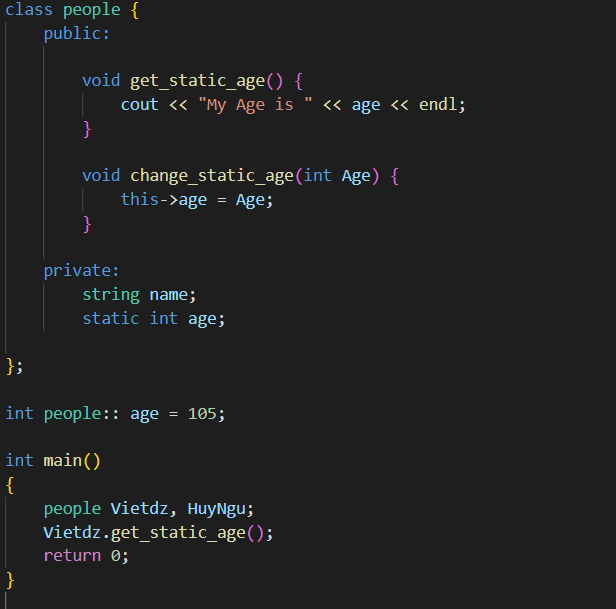
[2.1 Friend Class: 6](#_Toc111710620)

# Static and Friend trong OOP

## I. Static trong OOP

### 1.1 Static variables trong OOP

- Một biến Static được chia sẻ bởi tất cả các object của class. Ta bắt buộc phải gán giá trị cho 1 biến static bên ngoài class.

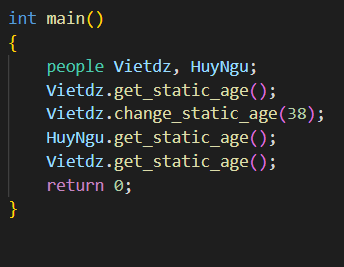


Kết quả:

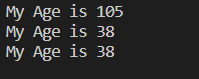


**Chú ý:** Biến static được chia sẻ bởi mọi đối tượng khác của class, nghĩa là khi thay đổi giá trị của biến static của 1 đối tượng thì giá trị của biến static của các đối tượng cũng thay đổi theo.

VD:

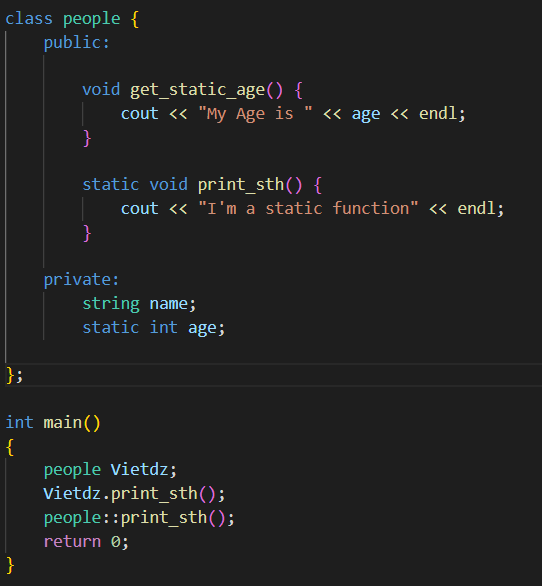


Kết quả:



### 1.2. Static function trong OOP

Bằng cách khai báo một function static trong class, Ta làm cho hàm đó không phụ thuộc vào một object nào của hàm. Một static function có thể được gọi ngay cả khi không có object nào của lớp tồn tại và các static function được truy cập chỉ bằng cách sử dụng tên lớp và toán tử phân giải phạm vi ::.

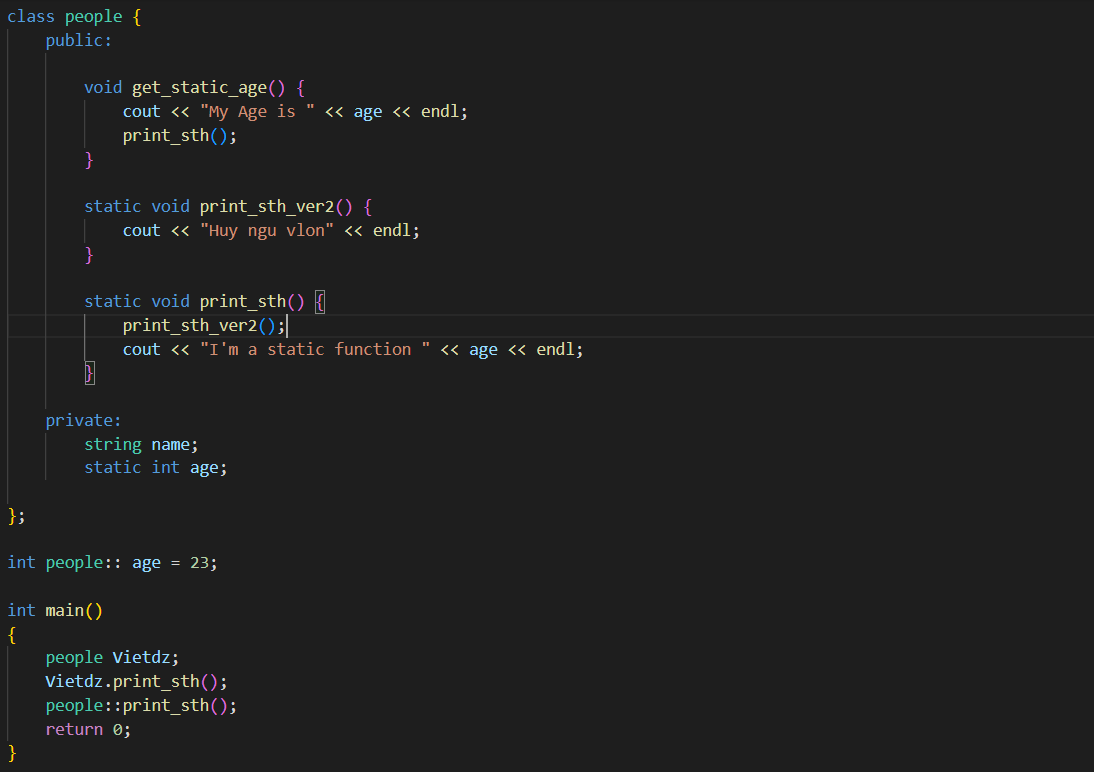


Kết quả:

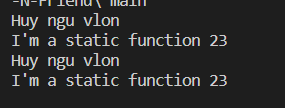


Các statcic function có không có quyền truy cập vào con trỏ this của lớp, cũng như các biến không phải static của Class. Các static function chỉ có thể call static function khác hoặc sử dụng các biến static.

Các phương thức của class có thể gọi được các static function, nhưng các static function không thể gọi các phương thức không phải static của class.



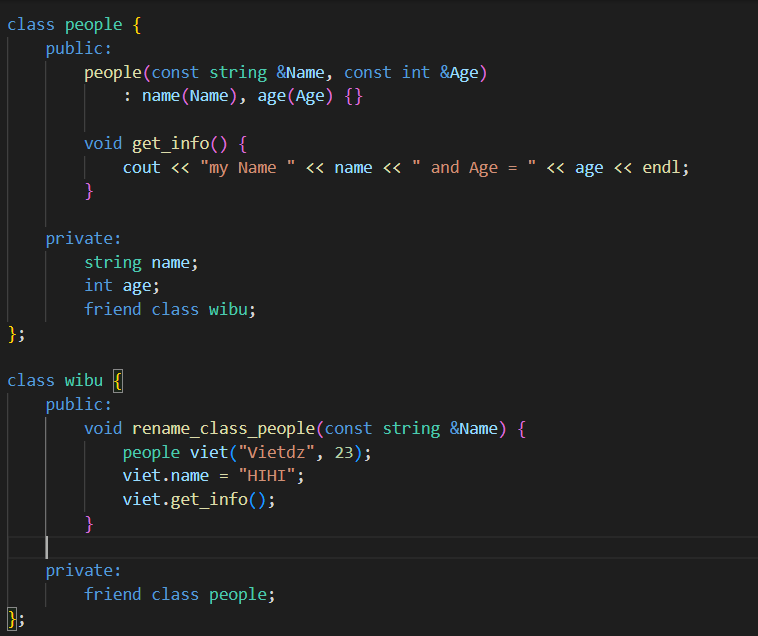
Kết quả:

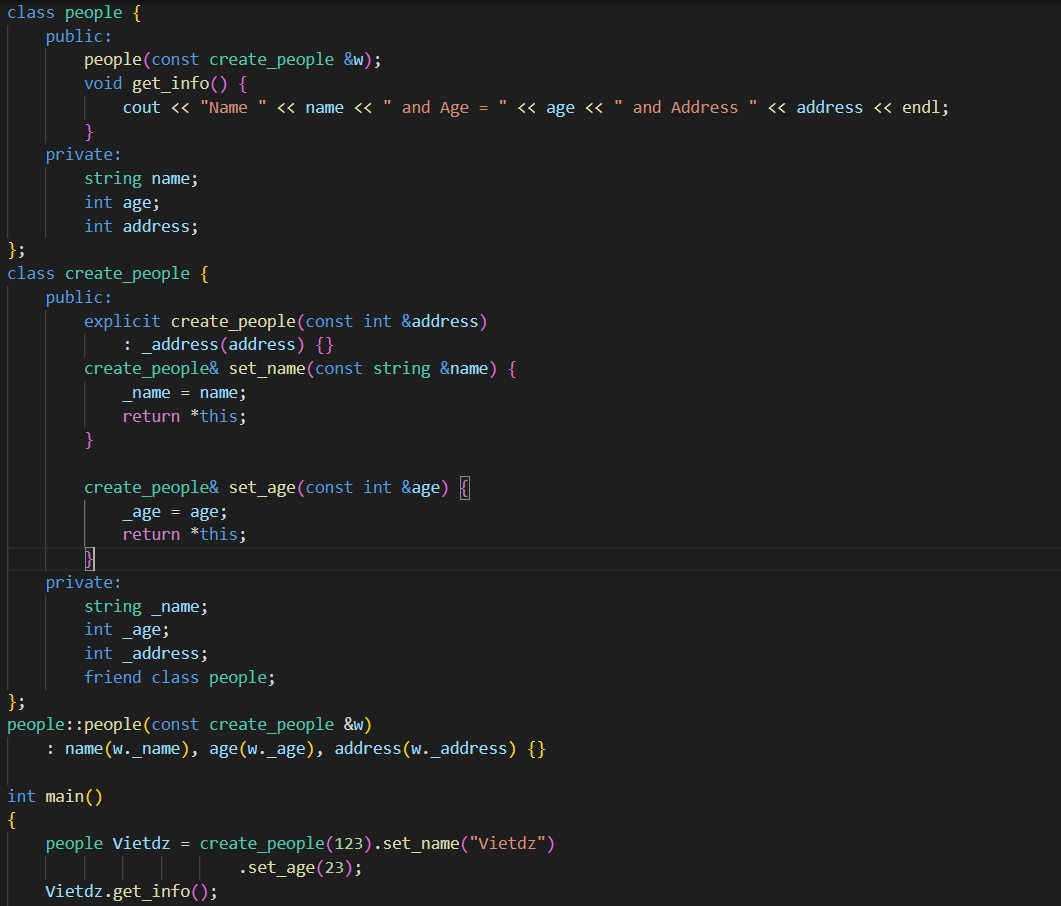


## II. Từ khóa friend trong OOP

### 2.1 Friend Class:

**Friend Class** Một friend class có thể truy cập các thành viên private và protected của lớp khác mà lớp đó được khai báo là friend.

Ta thấy biến viet ở class wibu có thể gọi trực tiếp các thuộc tính name và age của class people.

Đọc ví dụ để hiểu: 

Kết quả:

