Static members in a class are elements (fields, methods, or nested classes) that belong to the class itself rather than to instances of the class. This means there is only one instance of the static member shared among all instances of the class.

Static members example in C++ language:

C++

```
class MyClass {
public:
    // Static variable
    static int staticVariable;

    // Static method
    static void staticMethod() {
        cout<<"EasyExamNotes.com"
    }
};

// Initialize the static variable (mandatory in C++)
int MyClass::staticVariable = 0;
```

In C++, you access static members using the class name or through an instance of the class:

C++

```
// Accessing static variable
int value = MyClass::staticVariable;

// Invoking static method
MyClass::staticMethod();
```

```
// Alternatively, through an instance (not recommended for static
members)
MyClass obj;
int valueFromInstance = obj.staticVariable;
obj.staticMethod();
```

Related posts:

1. Abstraction and encapsulation
2. Object Oriented Programming & Methodolog Viva Voce
3. How to install compiler for code blocks
4. Object Oriented Programming
5. Differences between Procedural and Object Oriented Programming
6. Features of Object Oriented Paradigm
7. Inheritance in Object Oriented Programming
8. Object Oriented Programming
9. Introduction to Object Oriented Thinking & Object Oriented Programming
10. Difference Between Object-Oriented Programming (OOP) and Procedural Programming
11. features of Object oriented paradigm
12. Merits and demerits of Object Oriented methodology
13. Concept of Objects: State, Behavior & Identity of an object
14. Access modifiers
15. Instances in OOP
16. Message Passing in OOP
17. Construction and destruction of Objects