

Q1.What is cascading of input and output operators? Explain with suitable examples.

Ans:The cascading is a way to extract/insert multiple values from/into more than one variable using one cin/cout statement. The multiple use of << or >> in one statement is known of cascading. 1.

Cascading of output operator (>>): cout << " Hello " << " ISC "; cout << "Value of B=" << b; 2. Cascading of input operator (>>): int n1, n2, n3; cin >> n1 » n2 >> n3; cin >> n1 >> n2;

ACCESS MODIFIER

There are 3 types of access modifiers available in C++:

- **Public**

- **Private**
- **Protected**

Note: If we do not specify any access modifiers for the members inside the class, then by default the access modifier for the members will be **Private**.

1. Public: All the class members declared under the public specifier will be available to everyone. The data members and member functions declared as public can be accessed by other classes and functions too. The public members of a class can be accessed from anywhere in the program using the direct member access operator (.) with the object of that class.

2. Private: The class members declared

as *private* can be accessed only by the member functions inside the class. They are not allowed to be accessed directly by any object or function outside the class. Only the member functions or the friend functions are allowed to access the private data members of the class.

3. Protected: The protected access modifier is similar to the private access modifier in the sense that it can't be accessed outside of its class unless with the help of a friend class. The difference is that the class members declared as Protected can be accessed by any subclass (derived class) of that class as well.

C++ Identifiers

All C++ **variables** must be **identified** with **unique names**.

These unique names are called **identifiers**.

Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).

Note: It is recommended to use descriptive names in order to create understandable and maintainable code:

The general rules for naming variables are:

- Names can contain letters, digits and underscores
- Names must begin with a letter or an underscore (`_`)

- Names are case sensitive (myVar and myvar are different variables)
- Names cannot contain whitespaces or special characters like !, #, %, etc.
- Reserved words (like C++ keywords, such as int) cannot be used as names