Access Specifiers

Overview

The access modifiers specify the accessibility of a variable, method, constructor, or class. There are three types of access modifiers available in C++.

- Private
- Public
- Protected

i) Private: The private access modifiers are only accessible within the same class in which they are declared. The data members and methods declared as private cannot be accessible from outside the class. If we try to access the data members and methods, which are declared private from outside the class, the compiler will give you a compile-time error. Let's look at the example below.

Example:

```
#include<iostream>
using namespace std;
class Circle {
   // private data member
   private:
      double radius;
   // public member function
   public:
      double compute_area() { // member function can access private
         // data member radius
         return 3.14 * radius * radius;
      }
};
int main() {
```

```
    // creating object of the class
    Circle obj;
    // trying to access private data member
    // directly outside the class
    obj.radius = 1.5;
    cout << "Area is:" << obj.compute_area();
    return 0;
}
```

Output:

```
In
function 'int main()':
11: 16: error: 'double Circle::radius'
is private
double radius; ^
31: 9: error: within this context
obj.radius = 1.5; ^
```

ii) Public: The data members, class, and methods declared as public can be accessed from anywhere. This access modifier can be accessed within the class and outside the class using the direct member access operator (.) with that class's object. Let's look at the example.

Example:

```
#include<iostream>
using namespace std;
class Circle {
    public:
        double radius;
    double compute_area() {
        return 3.14 * radius * radius;
    }
};
// main function
int main() {
    Circle obj;
```

```
    // accessing public data member outside class
    obj.radius = 5.5;
    cout << "Radius is: " << obj.radius << "\n";
    cout << "Area is: " << obj.compute_area();
    return 0;
}
```

Output:

Radius is: 5.5

Area is: 94.985

iii) Protected: Protected access modifier is similar to private access modifier, i.e., it can't be accessed outside of its class unless, with the help of 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.

Example:

```
#include <iostream>
using namespace std;
class Parent {
   // protected data members
   protected:
      int id_protected;
};
class Child: public Parent {
   public: void setId(int id) {
      // Child class can access the inherited
      // protected data members of the base class
      id_protected = id;
   }
   void displayId() {
      cout << "id_protected is: " << id_protected << endl;
   }
};
int main() {
   Child obj1;
```

```
    // member function of the derived class can
    // access the protected data members of the base class
    obj1.setId(81);
    obj1.displayId();
    return 0;
}
```

Output:

```
id_protected is: 81
```

Difference between private, protected, and public modifiers:

Let's look at the summary of private, protected, and public access modifiers.

Visibility	Private	Protected	Public
Within the same class	Yes	Yes	Yes
In derived Class	No	Yes	Yes
Outside the class	No	No	Yes