Access Specifier

Access Specifiers are used to implement data hiding which is a very important aspect of object-oriented programming. Access Specifiers in a class are used to assign the accessibility to the class members, that is, they are used to set some level of restrictions on the class members as to not get directly accessible from outside the function. There are 3 types of access specifiers:

Public

- All the class members declared under this are available to everyone, that is, they can be accessed by other classes and functions too. The public members are directly accessible using the dot operator with the object of the class from anywhere in the program.

Example

C++

```
#include<iostream>
using namespace std;

class student{
  public:
      int marks;

      int getPercentage()
      {
         return (marks*100)/100;
      }
};

int main(){

  student student1;
  student1.marks=92;
  cout<<"Percentage = "<<student1.getPercentage()<<endl;
}
```

Output

```
Percentage = 92
```

Private

- All the class members declared under private can be accessed only by the member functions inside the class, they can not be accessed by any object or function outside the class. The member functions and the friend functions are allowed to access them.
- If we don't specify any access modifiers for the members inside the class, then by default, the access modifier for members is set as Private.

Example

```
#include<iostream>
using namespace std;

class student{
   private:
      int marks;
    public:
      int getPercentage()
      {
         return (marks*100)/100;
      }
};

int main(){

  student student1;
  student1.marks=92;
  cout<<"Percentage = "<<student1.getPercentage()<<endl;
}
```

Output

```
18:13: error: 'marks' is a private member of 'student'
   student1.marks=92;
            ^
r-6:13: note: declared private here
        int marks;
            ^
```

Example

C++

```
#include<iostream>
using namespace std;

class student{
    private:
        int marks;
     public:
        int getPercentage(int m)
        {
            marks=m;
            return (marks*100)/100;
        }
};

int main(){

   student student1;
   cout<<"Percentage = "<<student1.getPercentage(92)<<endl;
}
```

Output

```
Percentage = 92
```

Protected

- All the data members which are declared under this cant be accessed outside of the class unless with the help of a friend class. The only difference between private and protected is that protected class members can not be accessed by any subclass of that class as well.

Example

C++

```
#include <bits/stdc++.h>
using namespace std;

// base class
class Database{
    // protected data members
    protected:
    int marks;
};

// sub class or derived class from public base class
class student : public Database
{
    public:
    void setMarks(int m)
    {
        marks = m;
    }

    void displayMarks()
    {
        cout << "Marks = " << marks << endl;
    }
};


// main function
int main() {

    student obj1;
    obj1.setMarks(81);
```

```
   obj1.displayMarks();

}
```

Output

```
Marks = 81
```