OOPS

Access Modifiers

One of the main features of object-oriented programming languages such as C++ is ***data hiding***.

Data hiding refers to restricting access to data members of a class. This is to prevent other functions and classes from tampering with the class data.

However, it is also important to make some member functions and member data ***accessible*** so that the hidden data can be manipulated indirectly.

The access modifiers of C++ allows us to determine which class members are accessible to other classes and functions, and which are not.

For example:

```
#include <bits/stdc++.h>
using namespace std;
class Course {
  private:
   int id;
   string name;

  public:
   void getUserCount() {
      // code
   }

   void enrollUser() {
      // code
   }
};

int main() {
   Course c;
   return 0;
}
```

Here, the variables ***id*** and ***name*** of the ***Course*** class are hidden using the ***private*** keyword, while the member functions are made accessible using the ***public*** keyword.

Types of C++ Access Modifiers

In C++, there are 3 access modifiers:

Public

Private

Protected

Public Access Modifier

The ***public*** keyword is used to create public members (data and functions).

The public members are accessible from any part of the program.

```
#include <bits/stdc++.h>
using namespace std;
class Course {
  public:
   int users = 100;
   int getUserCount() { return users; }
};
```

```
int main() {
   Course c;
   cout<<"Enter Number of users: ";
   cin >> c.users;
   cout << c.getUserCount();
   return 0;
}
```

Output

```
Enter Number of users: 3
3
```

In this program, we have created a class named Course, which contains a ***public*** variable age and a ***public*** function ***getUserCount()***

In ***main()***, we have created an object of the ***Course*** class named ***c***. We then access the ***public*** elements directly by using the codes ***c.users*** and ***c.getUserCount()***

Notice that the ***public*** elements are accessible from ***main()***. This is because ***public*** elements are accessible from all parts of the program.

Private Access Modifier

The ***private*** keyword is used to create private members (data and functions).

The private members can only be accessed from within the class.

However, friend classes and friend functions can access private members.

```
#include <bits/stdc++.h>
using namespace std;
class Course {
  private:
   int id;
   string name;
   int users = 7;

  public:
   int getUserCount() { return users; }
};

int main() {
   Course c;
   cout << c.getUserCount();
   return 0;
}
```

Output

```
7
```

In ***main()***, the object ***c*** cannot directly access the class variable ***users***.

```
// Compilation Error
cout << c.users;
```

Protected Access Modifier

The ***Protected*** keyword is used to create protected members (data and function).

The ***Protected*** members can be accessed within the class and from the ***derived*** class.

Derived Class and Inheritance are discussed further in this module.

```
#include <bits/stdc++.h>
using namespace std;
class Course {
  protected:
   int users = 12;
};

class Batch : public Course {
  public:
   int getUserCount() { return users; }
};

int main() {
   Batch b;
   cout << b.getUserCount();
   return 0;
}
```

Output

```
12
```

Here, ***Batch*** is an inherited class that is derived from ***Course***. The variable ***users*** is declared in ***Course*** with the ***protected*** keyword.

This means that ***Batch*** can access ***users*** since ***Course*** is its parent class.

Summary

public: elements can be accessed by ***all*** other classes and functions.

private: elements cannot be accessed outside the class in which they are declared, except by ***friend*** classes and functions.

protected: elements are just like the private, except they can be accessed by ***derived*** classes.

Specifiers	Same Class	Derived Class	Outside Class
public	**Yes**	**Yes**	**Yes**
private	**Yes**	**No**	**No**
protected	**Yes**	**Yes**	**No**

Note: By default, class members in C++ are ***private***, unless specified otherwise.