



Lesson 30 CPP Inheritance

One of the most important concepts in **object-oriented programming** is that of inheritance. **Inheritance** allows us to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to **reuse** the code functionality and fast implementation time.

When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the **base class**, and the **new class** is referred to as the **derived class**.

Inheritance is the process by which new classes called **derived classes** are created from existing classes called **base classes**.

The **derived classes** have all the features of the **base class** and the programmer **can choose** to add new features specific to the newly created derived class.

Access specifiers:

1. If a member or variables defined in a class is **private**, then they are accessible by members of the same class only and cannot be accessed from outside the class.
2. **Public** members and variables are accessible from outside the class.



3. **Protected** access specifier is a stage between private and public. If a member functions or variables defined in a class are protected, then they cannot be accessed from outside the class but can be accessed from the derived class.

Example :

```
class MyClass
```

```
{ public:
```

```
MyClass(void) { x=0; }
```

```
void f(int n1)
```

```
{ x= n1*5;}
```

```
void output(void) { cout<<x; }
```

```
private:
```

```
int x;
```

```
};
```

```
class sample: public MyClass
```

```
{ public:
```

```
sample(void) { s1=0; }
```

```
void f1(int n1)
```

```
{ s1=n1*10;}
```

```
void output(void)
```

```
{ MyClass::output(); cout << s1; }
```

```
private:
```

```
int s1;
```



```
};  
int main(void)  
{ sample s;  
  s.f(10);  
  s.output();  
  s.f1(20);  
  s.output();  
}
```

Output :

50

200