**CHAPTER 3**

**INHERITANCE**

\*\* Inheritance 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.

\*\*The mechanism of deriving a new class from an old class/previous written class in known as inheritance. Also known as “is a” or ”kind of” or ”is a kind of” relationship.

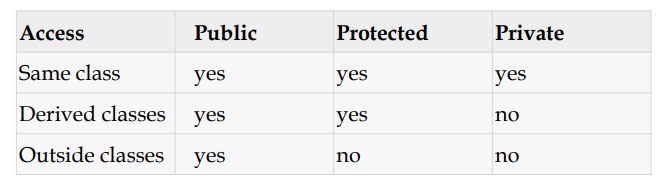
\*\*The class which is inherited is called base class/parent class/super class. The class that inherits the base class is known as sub class/child class/derived class.

class derived-class: access-specifier base-class

**Access Control in Inheritance**

\*\*A derived class can access all the non-private members of its base class. Thus baseclass members that should not be accessible to the member functions of derived classes should be declared private in the base class.

\*\*Default access specifier of inheritance from base class to child class is Private



**Access Control in Inheritance**

\*\* **Public base class:** public members of the base class are public members of the derived class. protected members of the base class are protected members of the derived class. private members of the base class remain private to the base class.

***[ Note by -Jannatul Ferdous Umama(Bristy)]***

\*\***Protected base class:** Both public and protected members of the base class are protected members of the derived class. private members of the base class remain private to the base class.

\*\* **Private base class:** Both public and protected members of the base class are private members of the derived class. private members of the base class remain private to the base class.

**Multiple Inheritance**

\*\* Deriving directly from more than one class is usually called multiple inheritance.

\*\* A class may inherit from more than one class by simply specifying more base classes, separated by commas, in the list of a class's base classes (i.e., after the colon).

\*\* For example, if the program had a specific class to print on screen called Output, and we wanted our classes Rectangle and Triangle to also inherit its members in addition to those of Polygon we could write:

class Rectangle: public Polygon, public Output;

class Triangle: public Polygon, public Output;

***[ Note by -Jannatul Ferdous Umama(Bristy)]***