**Inheritance**

This is the ability of a class to automatically access or inherit all the attributes and methods available in another class. The beauty of inheritance is that the aforementioned is done without having to retype all the attributes and methods to be inherited. When a class (class A) inherits the attributes and methods of another class (class B), class A is called the derived class while class B is known as the parent class. In this situation, class A would be able to access and modify all public and protected members of class B.

Inheritance is very important as it reduces code duplication to a large extent and makes our code readable and organized (as far as the chain of inheritance is reasonably short). For a class to inherit from another class (using c# as an example), the colon (:) sign is written in front of the class name and followed by the name of the class to be inherited. The example below demonstrates this.  
  
public class Person

{

private string \_name;

public Person(string name)

{

\_name = name;

}

public string GetName()

{

return \_name;

}

}

public class Student : Person

{

private \_class;

public Student(string name, string sClass) : base(name)

{

\_class = sClass;

}

public string GetClass()

{

return \_class;

}

}

Student newStudent = new Student(“Prince”, “Grade 5”);

string name = newStudent.GetName(); // output: Prince

string sClass = newStudent.GetClass(); // output: Grade 5

The above example demonstrates how the Student class inherits from the Person class and is able to invoke the Person class constructor and access the ‘GetName()’ method.