­**Abstraction**

Abstraction is the process of simplifying an idea or thing. This is usually done by hiding/removing the complex parts of the idea or thing and providing a simple or easy-to-use interface with which the purpose of the hidden complexities can be achieved. In abstracted systems, it is uncommon to see the implementation of things for instance, to turn on a car, you will need to turn in the key in the ignition, while this is a simple thing to do, the process that it takes for a car to actually turn on is complex and it involves a whole lot of things than just the key being twisted.   
  
In programming, creating classes is an example of abstraction. Classes allow us to identify important behaviors and states of something and there creating an easy way to define the thing. For example, a human being is a very complex entity, to create an abstraction of a human being, it would be important to identify attributes like firstname, lastname, age, sex, etc. Then, also identify actions that a human being can perform such as greeting, writing, etc. With this example, when asked to define a human, one can easily say a human being is something that has a name, and age; can talk, greet and even write. This removes a lot of complexities that have to do with a human being and makes it simple to understand.

A class that abstracts a human being can look like this:  
  
public class HumanBeing  
{

String firstname = “Prince”;

String lastname = “Chukwu”;

Int age = 24;

Public void Greet()  
 {

Console.Write($”Hello! My name is {firstname} {lastname}

}

}  
  
  
In summary, abstraction is the process of simplifying an entity.