**Learn Java programming tutorial lesson 8 - Inheritance**

**What is inheritance?**

Inheritance is the ability of objects in java to inherit properties and methods from other objects. When an object inherits from another object it can add its own properties and methods. The object that is being inherited from is called the parent or base class and the class that is inheriting from the parent is called the child or derived class.

**How to use inheritance**

We will first create a parent class called Person. We will then create a child class called Student. Then we will create a program to use the Student class. The following is a Person class which has a variable for the person's name. There are also set and get methods for the name.

class Person  
{  
   private String name;  
   
   public void setName(String n)  
   {  
      name = n;  
   }  
   
   public String getName()  
   {  
      return name;  
   }  
}

Now we will create the Student class that has a variable for the student number and the get and set methods for student number. The extends keyword is used to inherit from the Person class in the following example.

class Student extends Person  
{  
   private String stuNum;  
   
   public void setStuNum(String sn)  
   {  
      stuNum = sn;  
   }  
   
   public String getStuNum()  
   {  
      return stuNum;  
   }  
}

Finally we have to create a program that will instantiate a Student object, set the name and student number and then print the values using the get methods.

public class TestInheritance  
{  
   public static void main(String[] args)  
   {  
      Student stu = new Student();  
      stu.setName("John Smith");  
      stu.setStuNum("12345");  
      System.out.println("Student Name: " + stu.getName());  
      System.out.println("Student Number: " + stu.getStuNum());  
   }  
}

**Abstract inheritance**

Abstract classes are created only for the purpose of being inherited from. In fact you can't instantiate an object from an abstract class. You must put the abstract keyword in front of the class name declaration to create an abstract class. We can change the Person class above to show how it works.

abstract class Person  
{  
   private String name;  
   
   public void setName(String n)  
   {  
      name = n;  
   }  
   
   public String getName()  
   {  
      return name;  
   }  
}

**Interfaces**

Interfaces are like abstract classes because you can't instantiate them and they are only used for being inherited from. The difference is that interfaces are not allowed to have variables unless they are constants and methods are not allowed to have a body. The point of an interface is for you to override the methods that are declared in it. You must use the implements keyword instead of extends when using interfaces. Here is an example of the Person class as an interface and the Student class that uses the Person interface.

interface Person  
{  
   public void setName(String n);  
   public String getName();  
}  
   
class Student implements Person  
{  
   private String stuNum;  
   private String name;  
   
   public void setStuNum(String sn)  
   {  
      stuNum = sn;  
   }  
   
   public String getStuNum()  
   {  
      return stuNum;  
   }  
   
   public void setName(String n)  
   {  
      name = n;  
   }  
   
   public String getName()  
   {  
      return name;  
   }  
}