Types and Objects

Each day we come into contact with thousands of objects. People tend classify the objects they interact with in order to help make better sense of these objects. Once an object is classified or given a type, a person will have certain ideas about what the object is, how it is used, what it might look like etc.

If we are told that a certain object is a food, right away we have certain expectations about what characteristics this object will have. We expect it to have a certain taste, texture, smell and other qualities associated with food.

We don’t expect to use food to transport us to school in the morning or to provide listening entertainment for us while we do our homework.

Typing or classifying objects makes understanding the objects much simpler.

Meet Yukon

Danielle tells us that Yukon will be coming to school today. I am a bit puzzled and nervous about how she intends to pull this off. Last time I looked at the map this particular land mass appeared quite a bit bigger than the school. Besides, even though it not expressedly written, the school must have some policy against the bringing of northern territories to class.

Danielle then goes on to tell us that ‘Yukon’ is the name of her dog. Immediately our brains dismiss the logistics of moving lakes and mountain ranges and we think of furry canines with four legs, wagging tails and ‘drool’.

Obviously there is a big difference between an object that is of a type dog and an object that is of type territory. We do not expect a dog to have a population or a capital city nor do we expect a territory to fetch a stick or play dead.

Objects of different types have different characteristics. In this course we will often refer to these charateristics as an object’s properties.

Type = Class

In Computer Science we use the word class instead of the word type. Instead of saying Danielle's pet 'Yukon' is of type dog, we say 'Yukon' is of class dog.

Dog is the *class*. 'Yukon' is a specific *object* of type/class dog.

We can not pet dog, or feed dog. We can feed a specific dog such as 'Yukon'.

Class vs Object

Having a class does not mean physical objects have been created.

We have certain ideas about what cake is, what it is made of, how it is made, what it is used for and such. Cake is a concept that we know and understand but it is not something that we can see, touch or eat.

A cake, in contrast is something that we can see and eat. Eg. The cake that mom just made sitting in the fridge topped with chocolate icing that came with a warning of dire consequences if we touch it before the party.

This distinction is crucial in understanding how classes are used in programming.

Object….cake as something you can eat.

Match the Objects with the Class

In the left column are a series of classes; on the right a list of objects.
Using lines match the objects with the class they are best categorized with.

Car	Toronto
Teacher	Mr. Wright
City	My Chevrolet Impala
Drink	The coffee on my desk

Properties

Properties are the characteristics or state of an object.
The properties that an object has is determined by the object's class(or type).

If an object is in a food class it is going to have different properties(taste,smell) than an object in a car class(speed, miles per gallon).

When thinking of classes and objects and their properties, it is helpful to think of two types of phrases: **is a** and **has a**

Lets look at Yukon the dog again. Yukon, incidentally is a Golden Retriever.

Yukon is the object
Dog is the class
and for this example, colour is the property we will look at.

Yukon **is a** dog
Yukon **has a** colour.

is a or has a

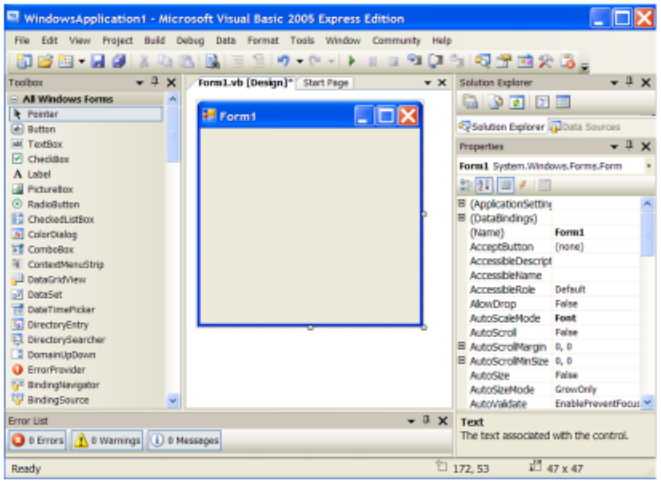
My Chevrolet Impala _____ car.

My Chevrolet Impala _____ colour.

My son Christopher _____ boy.

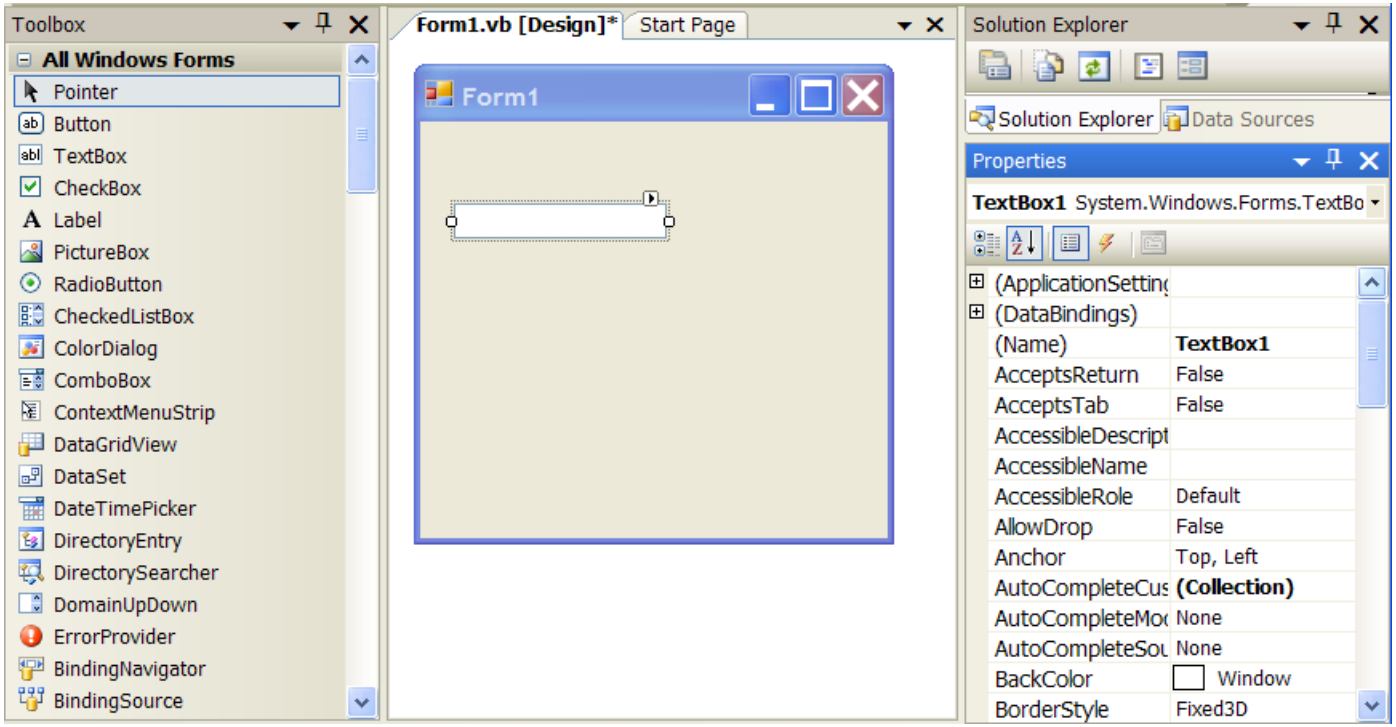
My son Christopher _____ height of about 40 inches.

The ToolBox



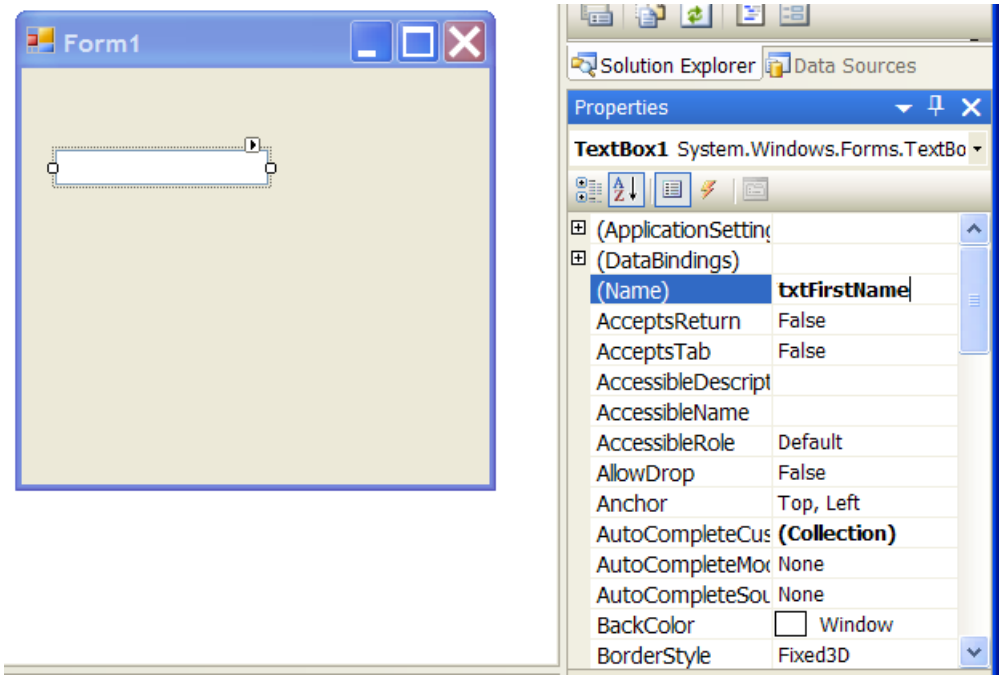
On the right hand side, above is a list of the items in the Toolbox. These various tools represent the classes that are available in Visual Basic. By choosing a tool and drawing these tools on the form, we create specific objects of that tool type/class.

Creating an Object



By choosing the TextBox tool and drawing a text box on the form, we are creating a specific instance or object of class TextBox. As soon as we draw this object on the form all of the properties associated with a text box show up on the right side in the Properties window. In the properties window, the name property tells us that this specific TextBox object will be called TextBox1.

Rename the Object



The value of the name property has been changed to txtFirstName

The txt prefix reminds us that the object is a text box and the rest of the name(using camel notation) gives us hints as to the purpose of the object.

TextBox1 was the default name that Visual Basic gave this object when it was created. It is a good idea to rename your objects in a clear and consistent manner so that you have a clear idea what this object is and how it will be used. This may seem unnecessary when you only have a couple of objects on a form but will become crucial as you try and keep track of objects in projects that are more complex.

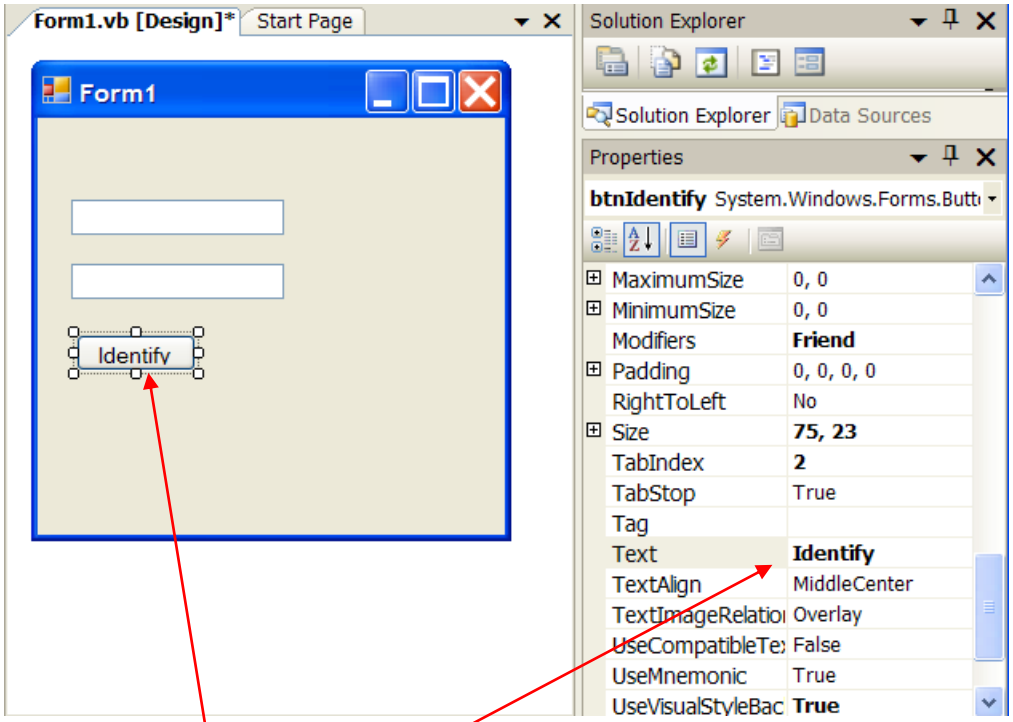
What is in a Name?

The name property is an object's most important property. The name is how we will refer to this specific object from now on.
Just as 'Yukon' refers to a specific object of class dog, txtFirstName refers to a specific TextBox object.

We can draw lots of other objects on the form but anytime we refer to txtFirstName we know which particular text box we are talking about.

Note* No two objects can have the same name. VB won't let you!!

Adding More Objects



btnIdentify has a text property value of “Identify”

In this example we have added another TextBox object called txtLastName and a Button object called btnIdentify.

Note that while the button is named btnIdentify its text property value is identify.

Do not cofuse an objects name property with its text property, they are two entirely different properties that affect the object in different ways.