Visual Basic for Kids

1. Introducing Visual Basic

A Story About Bill and Paul

Back in the early 1970's, Bill and Paul were friends at Lakeside School in Seattle, Washington. Bill and Paul enjoyed working with computers and wrote many games they could play - games like Tic-Tac-Toe and moon landing games. These computer games were written in a computer language called **BASIC** (which stands for **B**eginner's **A**ll-Purpose **S**ymbolic **I**nstruction **C**ode). When the first small computer was developed in the mid 1970's, Bill and Paul had just started college. They had improved their computer skills by then and were able to tell that new computer how to use the BASIC language. They actually sold their version of the BASIC language to other people (it cost $350 and was stored on a cassette tape). They started making money selling BASIC and decided to start a company. You may have heard of their company - Microsoft! Bill is Bill Gates, who still runs Microsoft and is worth many billions of dollars, and Paul is Paul Allen, who no longer works for Microsoft, but has his own business ventures including computers and software, professional sports teams, and real estate development.

Yes, the first product sold by Microsoft was the BASIC computer language. Microsoft's name for their BASIC language product has changed over the years, having names like GW-BASIC, QuickBasic and QBasic, with the newest version

being called **Visual Basic**. Visual Basic is one of the easiest programming languages to learn. Yet, even though it is easy to learn and to use, Visual Basic can also be used to develop very powerful computer programs. In these notes, you will learn how to use Microsoft's Visual Basic to write your own computer programs. You may not become a billionaire like Bill and Paul, but hopefully you'll have some fun learning a very valuable skill.

Let's Get Started

Learning how to use Visual Basic to write a computer program (like learning anything new) involves many steps, many new terms, and many new skills. We will take it slow, describing each step, term, and skill in detail. Before starting, we assume you know how to do a few things:

- You should know how to start your computer and use the mouse.
- You should have a little knowledge on working with your operating system (Windows 95, 98, or NT).
- You should know how to resize and move windows around on the screen.
- You should know how to run an application on your computer by using the **Start Menu**.
- You should know how to fill in information in Windows that may pop up on the screen.
- You should know about folders and files and how to find them on your computer.
- You should know what file extensions are and how to identify them. For example, in a file named **Example.ext**, the three letters **ext** are called the extension.

You have probably used all of these skills if you've ever used a word processor, spreadsheet, or any other software on your computer. If you think you lack any of these skills, ask someone for help. They should be able to show you how to do them in just a few minutes. Actually, any time you feel stuck while trying to learn this material, never be afraid to ask someone for help. We were all beginner's at one time and people really like helping you learn.

Let's get going. And, as we said, we're going to take it slow. In this first class, we will learn how to get Visual Basic started on a computer, how to load a program (or project) into Visual Basic, how to run the program, how to stop the program, and how to exit from Visual Basic. It will be a good introduction to the many new things we will learn in the classes to come.
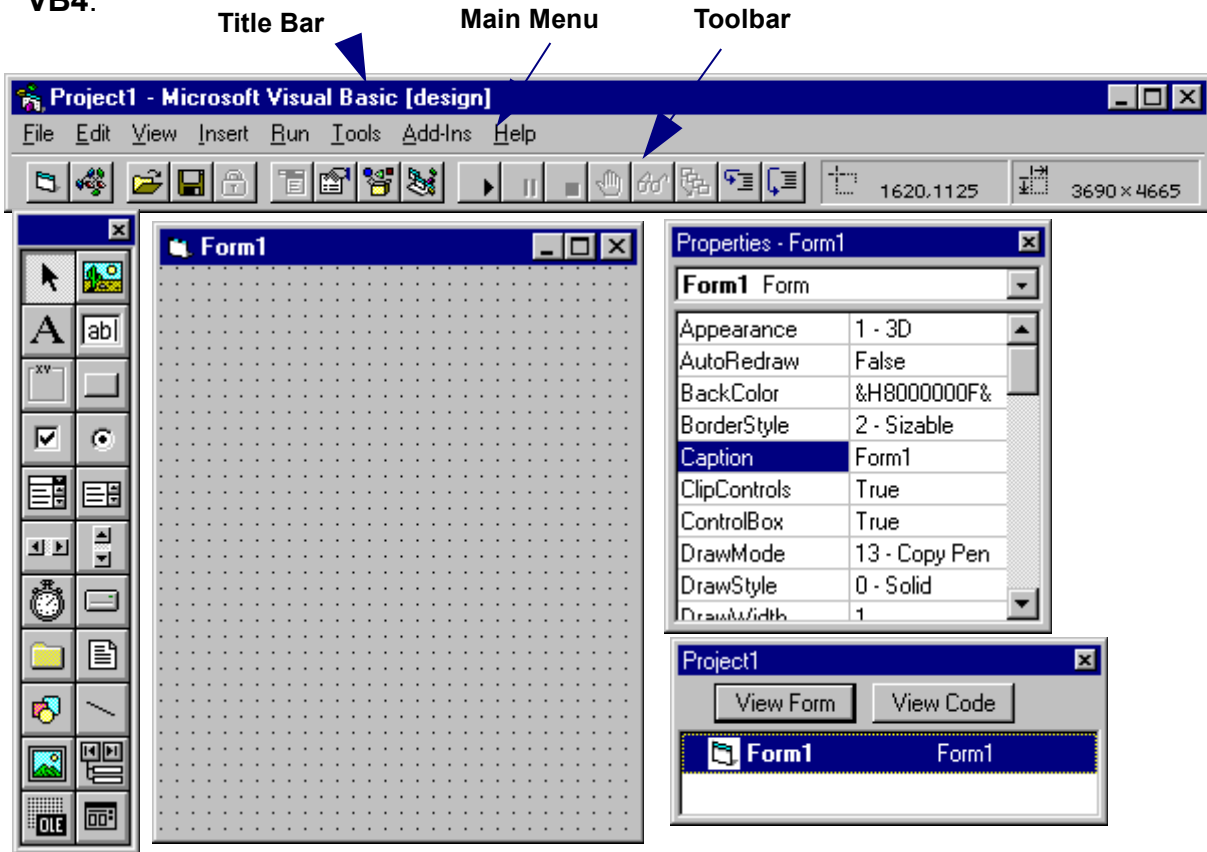
Starting Visual Basic

We assume you have some version of Visual Basic installed and operational on your computer. If you don't, you need to do this first. Again, this might be a good place to ask for someone's help if you need it. Over the past several years, there have been many versions of Visual Basic. You will be able to do this class using Visual Basic 4.0 (32 bit version), Visual Basic 5.0, or Visual Basic 6.0. Throughout the notes, we will refer to Visual Basic by its abbreviation **VB**, and you will be told what to do depending on what version of Visual Basic you are using. If you are using Visual Basic 4.0, look for the abbreviation **VB4** for information particular to your version. If you are using Visual Basic 5.0, look for the abbreviation **VB5** for information particular to your version. And, if you are using Visual Basic 6.0, look for the abbreviation **VB6** for information particular to your version.
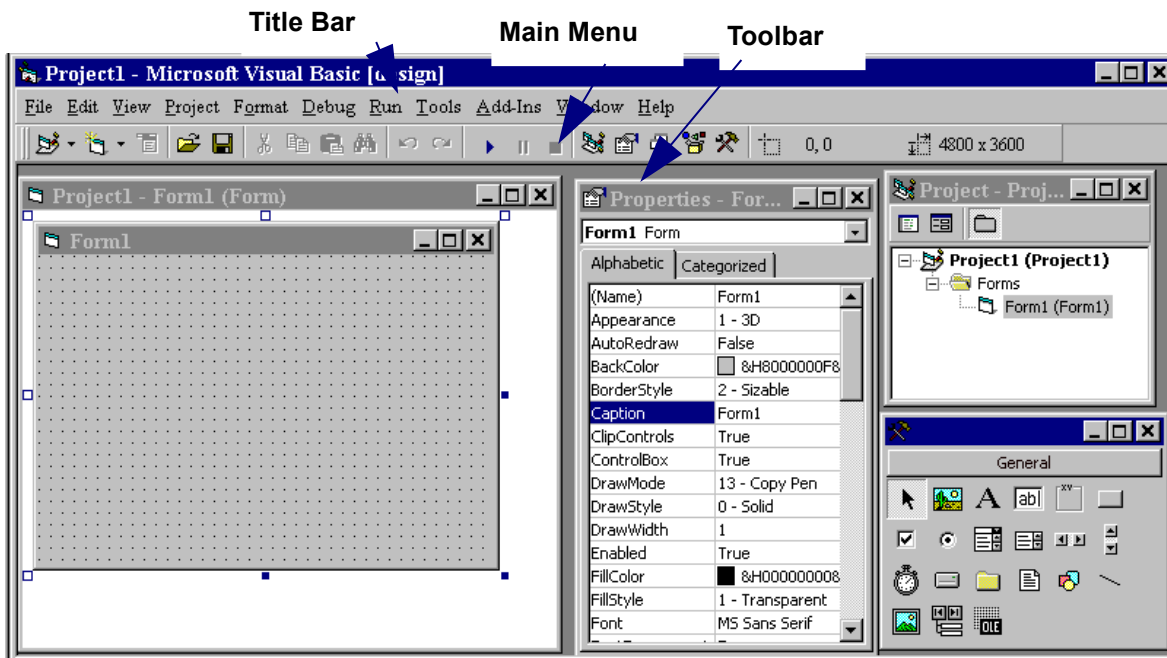
To start Visual Basic:

- Click on the **Start** button on the Windows task bar.
- Select **Programs**, then **Microsoft Visual Basic #** (where # is your version of Visual Basic).
- Click on **Visual Basic #** (where again # is your version).

(Some of the headings given here may differ slightly on your computer, but you should have no trouble finding the correct ones.) The Visual Basic program should start. Several windows will appear on the screen, with the layout depending on your version of VB.
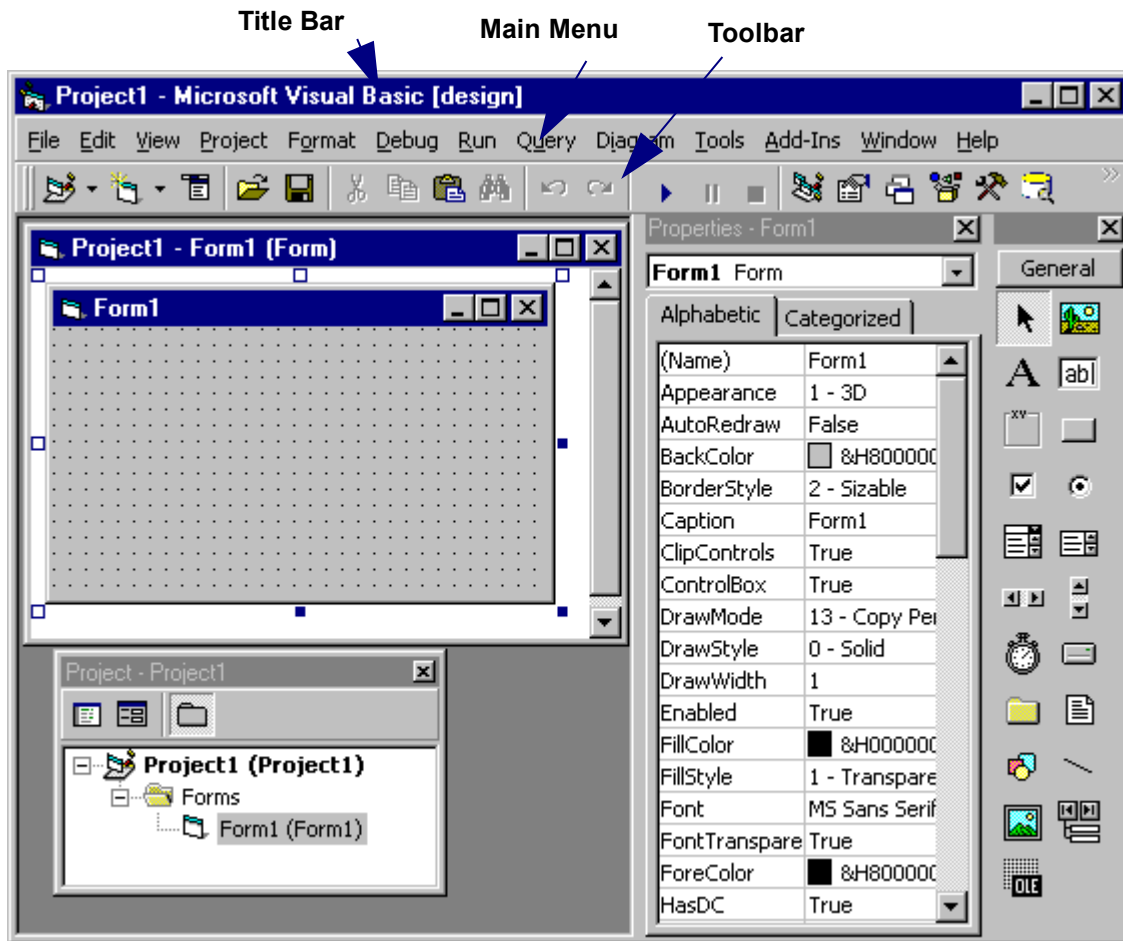
VB4:

Title Bar   Main Menu   Toolbar

VB5:

Title Bar   Main Menu   Toolbar

VB6:

Title Bar   Main Menu   Toolbar

When starting VB5 and VB6, a dialog box asking about **New Project** type may appear. If it does, just click the **Open** button.

      Let's point out just a few items on the screen. At the top of the screen is the Visual Basic **Main Window**. At the top of the main window is the **Title Bar**. The title bar gives us information about what program we're using and what Visual Basic program we are working with. Below the title bar is the **Main Menu** from where we can control the Visual Basic program. You should be familiar with how menus work from using other programs like word processors and games. Under the main menu is the **Toolbar**. Here little buttons with pictures also allow us to control Visual Basic, much like the main menu. If you put the mouse cursor over one of these buttons for a second or so, a little 'tooltip' will pop up and tell you

what that particular button does - try it! Almost all Windows applications (spreadsheets, word processors, games) have toolbars that help us do different tasks. This is the purpose of the Visual Basic toolbar. It will help us do most of our tasks. We'll worry about the other things on the screen later.

Opening a Visual Basic Project

What we want to do right now is **open a project**. Programs written using Visual Basic are referred to as **projects**. Projects include all the information we need for our computer program. When we are done writing our computer program using Visual Basic, we save it as a project. Included with these notes are many Visual Basic projects you can open and use. Let's open one now.

We will use the Visual Basic **toolbar** to open a project. Look for a button on the toolbar that looks like a folder opening up. This is the **Open Project** button:

- Click on this button. An **Open Project** window will appear.
- Find the folder named **VBKids** (stands for **V**isual **B**asic for **Kids**). This is the folder that holds the notes and projects for this course. Open that folder.
- Find the folder named **VBKProjects.** This folder holds all the projects for the course. In this folder are three other folders: **VB4Projects** (contains projects for Visual Basic 4.0, 32 bit version), **VB5Projects** (contains projects for Visual Basic 5.0), and **VB6Projects** (contains projects for Visual Basic 6.0). Select the folder for your version of Visual Basic.

Remember how you got to this folder. Throughout the course, you will go to this folder to open projects you will need.

Now, in your project file, find a project file named **Sample**. Open **Sample**. Many windows are now on the screen. Look for a window that looks something like this - this is **Sample**:
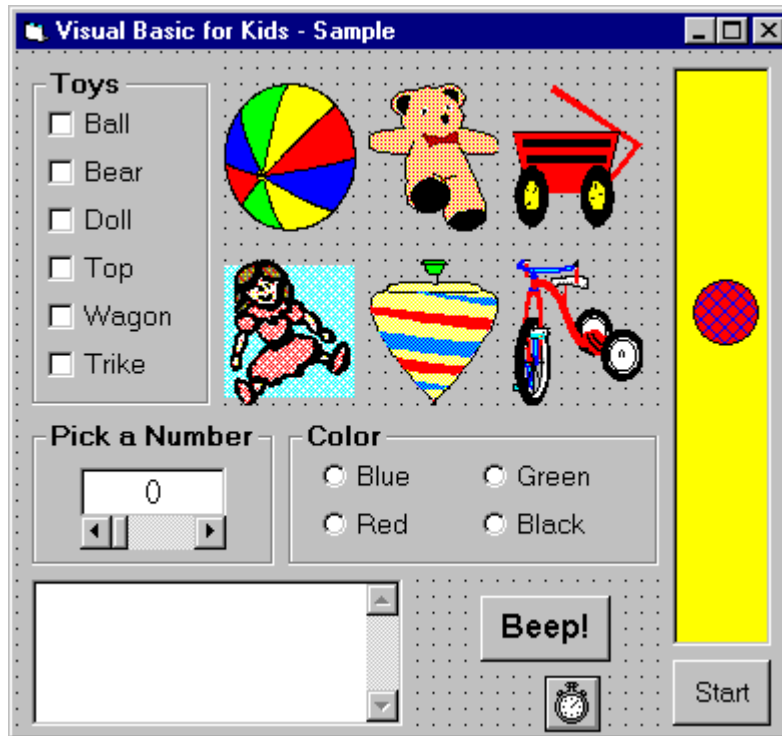
If this window doesn't show up when you open the project:

> **VB4**:   Click **View** on the main menu, then **Form**.
> **VB5, VB6**:   Click **View** on the main menu, then **Object**.

As an alternate, if the window does not show up, press the **F7** function key while holding down <**Shift**>. This is a keyboard command to show this particular window. We're going to spend a bit of time explaining everything that is displayed here. This will introduce you to some of the words, or vocabulary, we use in Visual Basic. There are lots of terms used in Visual Basic. Don't try to memorize everything - you'll see these new words many times through the course.

We call the displayed project window a **Form**. All Visual Basic projects or programs are built using forms. In fact, you have probably noticed that all Windows applications are built using forms of some type. At the top of the form is the **Title Bar**. It has an **icon** (little picture) related to the form, a description of what the form does (**Visual Basic for Kids - Sample**), and three smaller buttons that control form appearance (we won't worry about these buttons right now). There are lots of other things on the form. These other things are the 'heart' of a Visual Basic computer program.

You see a set of square buttons with toy names next to them. You see pictures of toys. You see a set of round buttons with color names next to them. There is a little box you can type in with something called a scroll bar on the right side. There's a big button that says Beep! There's a little device for picking the value of a number. And, there's a ball in a big rectangle with a button that says Start and a little thing that looks like a stopwatch. We call all of these other things on the form **Controls** or **Objects**. Controls provide an **interface**, or line of communication, between you (or the user of your program) and the computer. You use the controls to tell the computer certain things. The computer then uses what it is told to determine some results and displays those results back to you through controls. By the way, the form itself is a control. If you've used any Windows applications, you've seen controls before - you probably just didn't know they were called controls. As examples, buttons on toolbars are controls, scroll bars to move through word processor documents are controls, menu items are controls, and the buttons you click on when opening and saving files are controls.

I think you get the idea that controls are a very important part of Visual Basic, and you're right. They are the most important part of Visual Basic - they are what allow you to build your applications. We will spend much of this course just learning about controls. Right now, though, let's run this program and get some insight into how a Visual Basic project (and its controls) works. Before we leave though, look at the title bar in the Visual Basic main window - notice the

bracketed word **[design]**. This tells us that Visual Basic is in **Design** mode. That is the mode used to build a Visual Basic project. Always look at this title bar to see what mode Visual Basic is working in.

Running a Visual Basic Project

After developing a Visual Basic project, you want to start or run the program. This gets the program going and lets the user interact with the **controls** on the form and have the computer do its assigned tasks. We can run a project using the toolbar. Look for a button that looks like the **Play** button on a VCR, CD player, or cassette tape player:

▶

* Click this button to run **Sample** (the project we opened previously).

The project form will appear and look something like this:
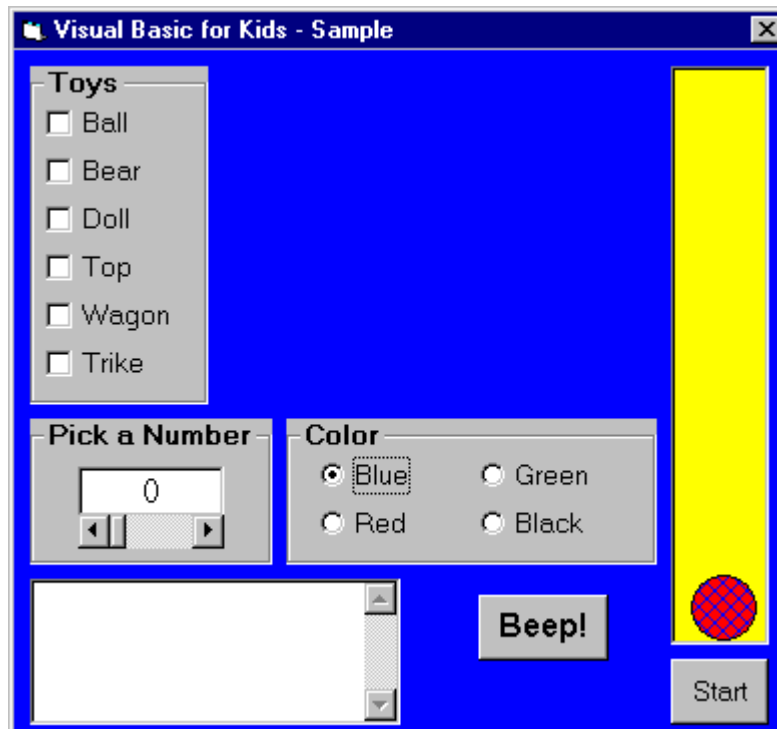
Notice a few things have changed. All the toys have disappeared. The background color of the form is blue. The circle button next to **Blue** has a black dot in it. The little stopwatch control has disappeared. The little ball has dropped to the bottom of the big rectangle. What happened? We'll find out how and why all this happened as we learn more about Visual Basic. Also, notice (in the main window) that Visual Basic is now in **Run** mode.

The project is now running, but what is it doing? Nothing is happening, or is it? At this point, Visual Basic is waiting for you, the user, to do something. We say your Visual Basic project is waiting for an **event** to occur. Nothing can happen in a Visual Basic program until an event occurs. We call Visual Basic an **event-driven** programming language. So, let's cause an event.

An event occurs when you do something on the form - click on something with the mouse, type something in places where words can go, or maybe drag an object across the form. In the upper left corner of the form is a group of six boxes within a rectangular region with the heading **Toys**. Each little box has a toy name printed next to it. Click on one of these boxes. Notice what happens. A check appears in the selected box, indicating box selection, and the toy named by that box appears on the screen. When we click on a box, we cause an event, called a **Click** event. The computer recognizes the event and does what you have told it to do (through your computer program) if that particular event occurs. In this case, the event tells the computer to display the selected toy. Click on the box again. The check mark and the toy disappear. You have caused another event and told the computer to make the toy disappear. This particular control is called a **check box**. Notice you can check as many boxes as you want, picking which toys (if any) you want displayed on your screen. Check boxes are used when you want to select items from a list. Two other controls are used in this example. The rectangular region the check boxes are contained is called a **frame**. The region each toy picture is displayed in is called an **image** control. Now, let's look at causing events with the other controls on the form.

Near the middle of the screen is a group of four round buttons in a frame with the heading **Color**. Each button has a color name printed next to it. The **Blue** button has a black dot in it, indicating it is the currently selected color (notice the form is blue). Click on another of these buttons. Notice what happens. The form color changes to the selected color. This **Click** event tells the computer to change the form background color. Notice that when you select a new color, the black dot appears in the selected button and disappears in the previously selected button. Unlike the check boxes we saw earlier, you can only select one of these buttons. This makes sense - the form can only be one color! These round buttons are called **option buttons**. Option buttons are used when you need to choose exactly one option from a list of many.

Under the **Toys** frame is another frame with the heading **Pick a Number**. There we see a number displayed in a control called a **label**. Under the label is another control called a **scroll bar**. You've probably seen scroll bars in other applications you have used. The scroll bar is used to change the displayed number. Click on the arrow on the right side of the scroll bar. The displayed value will increase by 1. Continued clicking on that arrow will continue to increase the value. Click on the white area to the left of the arrow. The value will increase by 10. Click on the little bar (called a thumb) in the middle of the scroll bar and drag it right, then left. Notice that, too, changes the displayed value. In this example, the computer is responding to the scroll bar's **Change** event, which occurs each time the scroll bar thumb moves. The Change event tells the computer that the scroll bar value has changed and allows the new value to be displayed in the label control.

Under the **Pick a Number** frame is a region with a scroll bar on the right side. This control is called a **text box**. You can click in it, then type in any text you want. Try it. The text box is like a little word processor in itself. Each time you type something in the text box, several events occur. There is a **KeyPress**

event when you press a key and a **Change** event that is called each time the text in the box changes.

Next to the text box is a button that says **Beep!** Click the button and you should hear a beep on your computer's speaker. This control is called a **command button** and is one of the most widely used controls in Visual Basic. The **Click** event told the computer to make the speaker beep.

The last thing on our form is a tall, yellow, rectangular control called a **picture box** that contains a little circle called a **shape** control. Under the picture box is a command button that says **Start**. Click on that button, that is, cause a **Click** event. The ball starts moving up. It continues moving up until it hits the top of the picture box, then starts moving back down. It will continue to do this until you click the command button that now says **Stop**. Remember the little stopwatch that was on our form in design mode, but disappeared when we ran the project. It is being used by the bouncing ball example - it is called a **timer** control. The Click event on the command button, in addition to changing what the button says to **Stop**, also started this timer control. The timer control generates events all by itself at preset time intervals. In this example, a **Timer** event is generated every 1/10th of a second and, in that event, the ball position is changed to give the appearance of movement. Notice that even while the ball is bouncing, you can change the form color, make animals appear and disappear, type text, and make the computer beep. So, Visual Basic even has the capability of handling multiple events.

Obviously, this project doesn't do much more than demonstrate what can be done with Visual Basic, but that is a important concept. It points out what you will be doing in building your own Visual Basic projects. A project is made up of the controls that let the user provide information to the computer. By causing events with these controls, the computer will generate any required results. We haven't worried about how to use the events to determine these results, but we

will in all the later classes. By the time you have finished this course, you will be able to build projects that do everything (and more) that the **Sample** project does. Let's look now at how to stop the project.

Stopping a Visual Basic Project

There are many ways to stop a Visual Basic project. We will use the toolbar. Look for a button that looks like the **Stop** button on a VCR, CD player, or cassette tape player (you may have to move the project form down a bit on the screen to see the toolbar):

*   Click on this button. The project will stop and Visual Basic will return to design mode.

An alternate way to stop the project is to use the **Close** button found on the form. It is the little button that looks like an **X** in the upper right corner of the form.

Stopping Visual Basic

When you are done working with a Visual Basic project, you want to leave the Visual Basic program. The procedure to leave Visual Basic is the same for all versions of Visual Basic. In fact, it is the same procedure used by nearly all Windows applications:

- Select **File** in the main menu.
- Select **Exit** (at the end of the File menu).

Stop Visual Basic now. Visual Basic will close all open windows and you will be returned to the Windows desktop. In stopping Visual Basic with **Sample** active, you may be asked if you want to save certain files. Answer **No**. Like with stopping a project, an alternate way to stop Visual Basic is to click on the close button in the upper right hand corner of the main window. It's the button that looks like an **X**.

Summary

Whew! Are you tired? We covered a lot of new material here, so if you are, that's OK. As we said earlier, you learned a lot of new words and concepts. Don't worry if you don't remember everything we talked about here. You will see the material many times again. It's important that you just have some concept of what goes into a Visual Basic project and how it works. And you know how to start and stop Visual Basic itself.

In summary, we saw that a Visual Basic project is built upon a **form**. **Controls** are placed on the form that allow the user and computer to interact. The user generates **events** with the controls that allow the computer to do its job. In the next class, you will begin to acquire the skills that will allow you to begin building your own Visual Basic projects. You will see how the parts of a project fit together. Using project **Sample** as an example, you will learn how to locate important parts of a project. Then, in Class 3, you will actually build your first project!