



Computer Programming

SETON HOME STUDY SCHOOL

Lesson Plan ♦ Answer Keys ♦ Tests ♦ Quarter Report Forms



Course Manual

CMP414_19A

OBJECTIVE

To understand conditional logic.

Work through Chapter 4 in the Small Basic manual. Conditional logic is probably the most fundamental part of computer programming. In almost every program, there are conditions, and one pathway through the program will be chosen based upon these conditions.

Note that the code from an If condition through to the EndIf is called a *code block*. The If...EndIf pair is called a *structure*. You will learn about many other code structures in the coming lessons.

The manual does not mention it, but an If ... EndIf block can also include ElseIf. You can use ElseIf when there is more than one possible condition. For example, you might want to categorize people by age as adults, teens, or children. You would do that like this:

```
If age < 13 then
    TextWindow.WriteLine("You are a child.")
ElseIf age < 20 then
    TextWindow.WriteLine("You are a teen.")
Else
    TextWindow.WriteLine("You are an adult.")
EndIf
```

When you use ElseIf, the program runs the first line which is true and then goes to the end of the block. So, if you enter an age of 8 for the block above, it would write the line saying the user is a child. Even though 8 is less than 20, the program will not execute the teen line since it only executes one of the conditional lines.

LESSON 4 QUIZ

1. What is the word that you would put in between two blocks of conditional code?

2. Complete the follow program:

```
TextWindow._____("How old are you? ")
UserAge=TextWindow.ReadNumber()
If UserAge<18 _____
    TextWindow.WriteLine("You are a child.")
_____
    TextWindow.WriteLine("You are an adult.")
EndIf
```

Activity

Write a program which accepts two numbers from a user and then reports whether the first number is evenly divisible by the second number. For example, if the user enters 20 and then enters 3, the program would report:

```
20 is not evenly divisible by 3
```

If the user enters 20 and then enters 2, the program would report:

```
20 is evenly divisible by 2
```

OBJECTIVE

To learn about message boxes and input boxes.

Work through Hour 17, from page 391 through page 410. Message boxes and input boxes are very helpful and useful since they allow you to either tell the user something or get input from the user without having to create a whole new form.

In the section about interacting with the keyboard and mouse events, note that it's pretty uncommon to use most of those events. For the mouse, the Click event is used almost all the time, and for the keyboard you would probably never use anything but the KeyPress event.

Besides handling keyboard input at the level of a particular control, as the book talks about, you can also handle keyboard input at the form level. Suppose that you had multiple textboxes, spinners, buttons, etc., on a form and you wanted to perform some action, such as printing, when the user pressed Control-P. You could add code to every control on your form, but it would be better to add a form KeyPress event, which would catch the event no matter what particular control was active.

In order to catch a form KeyPress event, you need to set the form's KeyPreview property to True. Then, you need to set your handling code in the form's KeyPress (or KeyDown or KeyUp) event.

LESSON 32 QUIZ

1. Name the five different icons that can be displayed in a MessageBox.

_____	_____
_____	_____

2. If a user presses the Cancel button of a InputBox, what value is returned to the program?

3. In order to catch a KeyPress event at the form level, what form property must be set to True?

Activity

Do Exercise 2 on page 411 of your book.

Updated 2/28/2019



CMP41419ALP