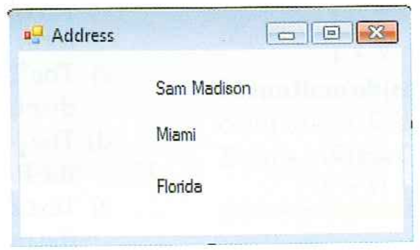
Exercises

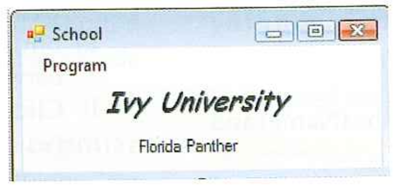
Exercise 1 — Address

Create an Address application that displays your name, city, and state in three separate labels. The interface should look similar to:



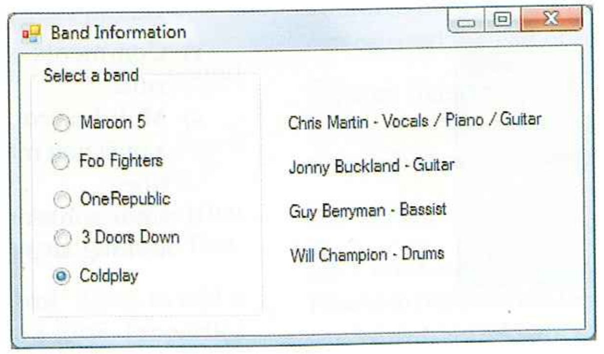
Exercise 2 — School

Create a School application that displays your school's name and mascot centered in two separate labels. Choose a different font, style, and size for the school name. Include a Program menu with an Exit command. The interface should look similar to:



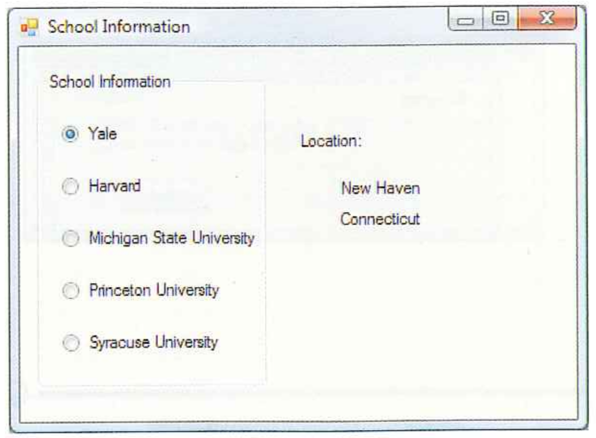
Exercise 3 — Band

Create a Band application that displays the members of a selected band. Include at least three of your favorite bands. The interface should look similar to:



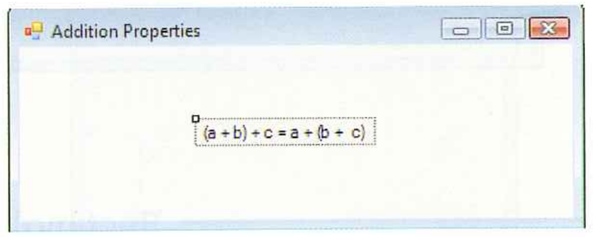
Exercise 4 — SchoolInformation

Create a SchoolInformation application that displays the city and state of a selected school. Include at least five of your favorite schools. The interface should look similar to:

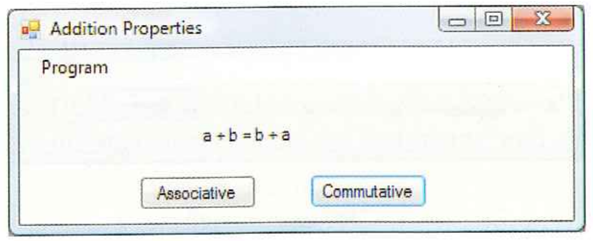


Exercise 5 — AdditionProperties

a) Create an AdditionProperties application that shows the associative property of addition, (a + b) + c = a + (b + c), in a label. The interface should look similar to:



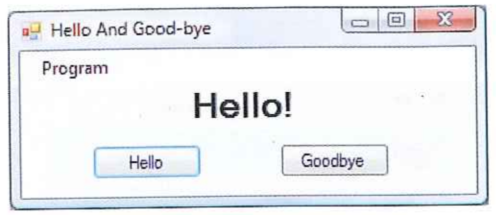
b) Modify the AdditionProperties application to display the associative property of addition when one button is clicked and the commutative property, a + b = b + a, when another button is clicked. The interface should look similar to the following after clicking Commutative:



c) Modify the AdditionProperties application to include a Program menu with Associative, Commutative, and Exit commands.

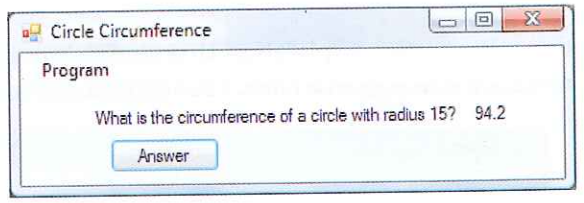
Exercise 6 — HelloAndGood-bye

Create a HelloAndGood-bye application that displays a label centered, bold, and size 18 that reads Hello! or Good-bye! depending on the button clicked. Include a Program menu with Hello, Good-bye, and Exit commands. The interface should look similar to the following after clicking Hello:



Exercise 7 — CircleCircumference

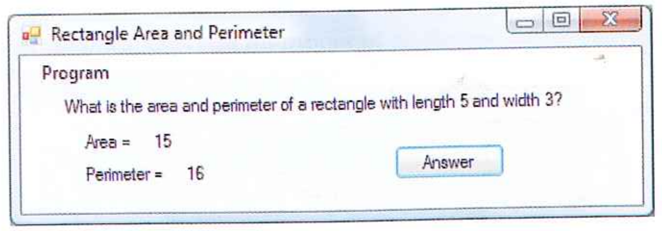
Create a CircleCircumference application that displays in a label the circumference ($2\pi r$) of a circle with radius 15. Use the value 3.14 for π. Include a Program menu with an Exit command. The interface should look similar to the following after clicking Answer:



What is the circumference of a circle with radius 15? 94.2

Exercise 8 — RectangleAreaAndPerimeter

Create a RectangleAreaAndPerimeter application that displays the area (length * width) and perimeter (2l + 2w) of a rectangle of length 5 and width 3. Include a Program menu with an Exit command. The interface should look similar to the following after clicking Answer:



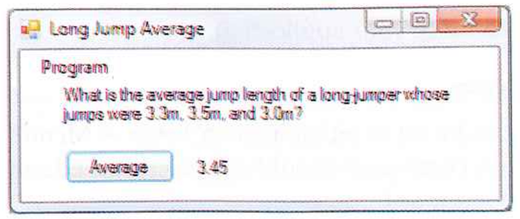
What is the area and perimeter of a rectangle with length 5 and width 3?

Area = 15

Perimeter = 16

Exercise 9 — LongJumpAverage

Create a LongJumpAverage application that calculates and displays the average jump length of an athlete whose jumps were 3.3m, 3.5m, 4.0m, and 3.0m. Include a Program menu with an Exit command. The interface should look similar to the following after clicking Average:

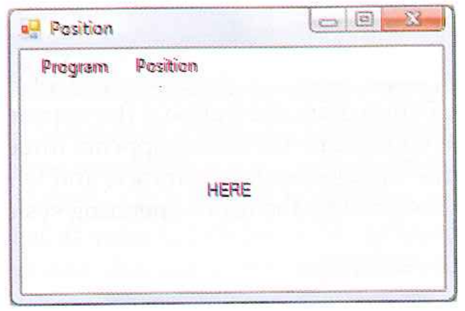


What is the average jump length of a long-jumper whose jumps were 3.3m, 3.5m, and 3.0m?

3.45

Exercise 10 — Position

Create a Position application that changes the position of text in a label according to the command selected by the user. The application should include a Program menu with an Exit command and a Position menu with TopLeft, TopCenter, TopRight, MiddleLeft, MiddleCenter, MiddleRight, BottomLeft, BottomCenter, and BottomRight commands. The interface should look similar to the following after selecting MiddleCenter from the Position menu:



Hint: Be sure the label is large enough to show changes in text position. Also, use the IntelliSense list for selecting the assignment value for the TextAlign property.

Exercise 11 — WithStatement

The `With` statement can be used to change several properties of a designated object within a single statement. For example, the following statement changes the text, sizing, and alignment of a label:

```
With Me.lblCity
   .Text = "Dallas"
   .AutoSize = False
   .TextAlign = ContentAlignment.MiddleCenter
End With
```

Create a WithStatement application that uses the With statement to change the properties of a label when a Change button is clicked. Have a classmate test your application.

Exercise 12 — HiBye

Create a HiBye application that displays "Hi" when the user clicks a Hi button or selects the Hi command from the File menu. The application should also include a Bye button and a Bye command, which ends the application when selected. The Bye command should be disabled until the user clicks either the Hi command or the Hi button. An Exit command should be included to quit the application. Use the properties listed below to enhance your application.

MenuItem properties:

- **Enabled** can be set to either True or False. A MenuItem that is not enabled appears dimmed. A command should appear dimmed when it should not be selected at that time.
- **ShortcutKeys** can be set to a keyboard shortcut. The keyboard accelerator is selected from the list in the ShortcutKey property.
- **ShowShortcutKey** can be set to True or False. When True, the keyboard accelerator selected in the ShortcutKey property is displayed along with the MenuItem text at run time.

A menu can also be modified to include bars that separate groups of commands. Right-click an existing menu item in the Design window to display a menu and then select Insert → Separator. A bar is inserted above the menu item. Right-click a bar and select Delete from the displayed menu to remove the separator.

RadioButton and Button properties:

- **FlatStyle** can be set to Flat, Popup, Standard, or System. Flat gives the button a flat appearance, Popup gives the button a flat appearance until the mouse moves over the button at which time the button appears three-dimensional, Standard gives the button a three-dimensional appearance, and System gives the button the default appearance assigned by the user's operating system.

Exercise 13 — SystemClock

The system clock properties can be used display the current time and date. The TimeString property sets or returns the current time from the system clock and the DateString property returns the current date. The Now property returns a value that represents the current date and time. Design and create a SystemClock application that displays the current time and time in a label. Share your application design with a classmate. *Note that this exercise will be updated in Chapter 9 with a Timer, allowing the display of a running clock.*

Variables and C

Key Concepts

Declaring and initializing variables and constants
Obtaining user input
Differentiating between data types
Understanding variable scope
Identifying Visual Basic keywords
Recognizing syntax and logic errors
Debugging an application
Creating an executable file
Application deployment
Problem-solving using a case study approach

Case Study

Calculator application

Declaring Variables

A *variable* is a name for a value stored in memory. Variables are used in programs so that values can be represented with meaningful names. For example, when a variable named `length` is used in a program, it is clear that its value is a distance. Variables should be used to represent values because they make code easier to read, understand, and modify.

declaration statement

A variable must be declared before it is used. A *declaration statement* takes the form:

```
Dim variableName As type
```

identifier
data type

A `Dim` statement must include the variable name, called the *identifier*, and the *data type*, which determines the type of data the variable will store. For example, the statement

```
Dim length As Integer
```

integer

declares a variable `length` to store data of type `Integer`. An *integer* is a numeric value that is a positive or negative whole number. When an `Integer` variable is declared it stores the value 0.

An identifier must begin with a letter and contain only letters, numbers, and some special characters. Typically variable identifiers begin with a lowercase letter. Any word after the first in a variable identifier should begin with an uppercase letter. For example, `rectangleLength`. This code convention allows variables to be easily recognized.

Multiple variables with the same data type can be declared in a single statement, similar to:

```
Dim length, width As Integer
```