

## True/False

17. Determine if each of the following is true or false. If false, explain why.
- a) A child control will always inherit property values from the parent control.
  - b) The ForeColor of an object is the color of its text.
  - c) The color of a form can be changed at run-time.
  - d) Visual Basic applications support audio files in WAV format.
  - e) A program can only have one Timer object.
  - f) A Click event executes at every timer interval.
  - g) If a timer should process an action every 10 seconds, then its Interval property should be set to 10.
  - h) The coordinate (0, 0) is the coordinate of the center of the object.
  - i) A line thickness must be indicated when declaring a SolidBrush object.
  - j) DrawCurve() would be the method to draw a pentagon.
  - k) A form Click event occurs when a form is clicked.
  - l) A Timer object can be turned on and off at run time by setting its Visible property.

## Exercises

### Exercise 1

### PhotoAlbum

- a) Create a PhotoAlbum application that allows the user to scroll through photographs and their corresponding captions by clicking a button that displays an image of an arrow. The arrow.jpg, anemone.jpg, grayangel.jpg, sponges.jpg, starfish.jpg, and scorpionfish.jpg images are data files for this text. The application interface should look similar to the following:

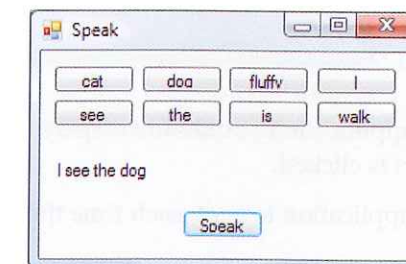


- b) Modify the PhotoAlbum application to play a bubble background sound while the application is running. Use the bubbles.wav file, which is a data file for this text.

### Exercise 2

### Speak

Create a Speak application that allows the build a string by clicking buttons on the interface. When the string is complete, the user can then click Speak to hear the string as a spoken phrase. The word-dog.wav, word-cat.wav, word-fluffy.wav, word-I.wav, word-is.wav, word-see.wav, word-the.wav, and word-walk.wav sound files are data files for this text. The application interface should look similar to:

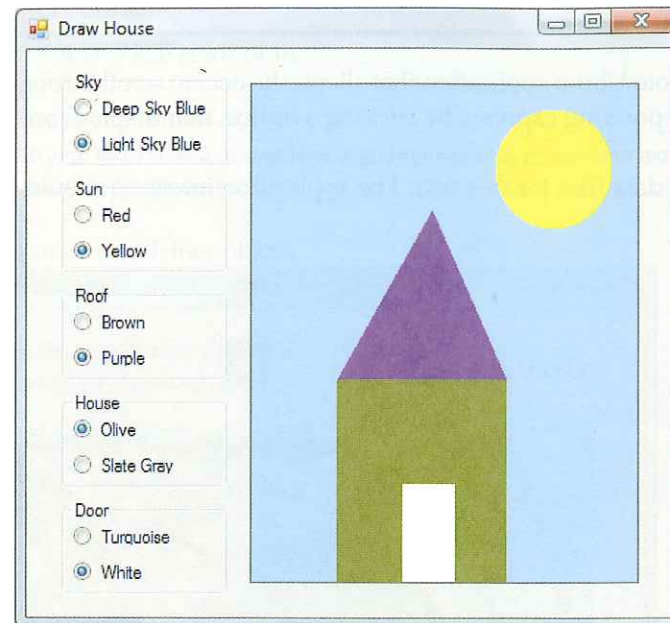




### Exercise 3

### DrawHouse

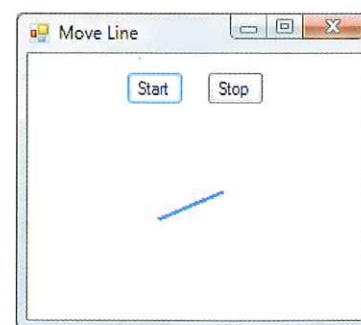
Create a DrawHouse application that allows the user to select the color of the sky, sun, roof, house, and door to draw in a label by clicking radio buttons. The application interface should look similar to:



### Exercise 4

### MoveLine

- Create a MoveLine application that displays a blue line that randomly changes position every 3 seconds after clicking Start. Clicking Stop should stop the animation. Hint: Use the size of the form and the Rnd() function to define starting and ending coordinates and the DrawLine() method to draw the lines. The form will need to be cleared before a new line is drawn. The application interface should look similar to the following after clicking Start:

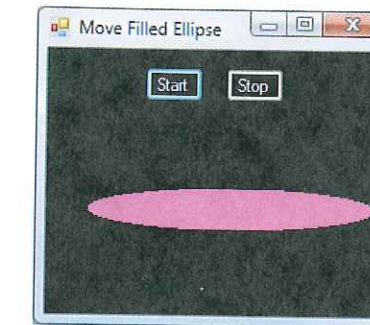


- Modify the MoveLine application to randomly display the moving line in either blue, green, or red when Start is clicked.
- Modify the MoveLine application to beep each time the line is displayed.

### Exercise 5

### MoveFilledEllipse

- Create a MoveFilledEllipse application that displays a filled ellipse that randomly changes position every 3 seconds after clicking Start. Clicking Stop should stop the animation. Hint: Use the size of the form and the Rnd() function to define coordinates and the FillEllipse() method to draw the ellipses. The form will need to be cleared before a new ellipse is drawn. The application interface should look similar to the following after clicking Start:

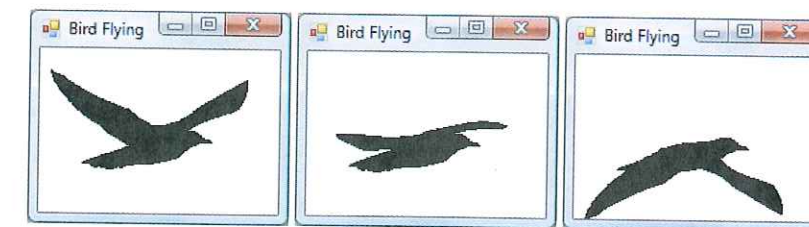


- Modify the MoveFilledEllipse application to randomly display the moving filled ellipse in either blue, green, or red when Start is clicked.

### Exercise 6

### BirdFlying

- Create a BirdFlying application that simulates a bird flying. Use the bird1.gif, bird2.gif, and bird3.gif images that are data files for this text. The three different animations should look similar to:



- Modify the BirdFlying application to include the sound of a flying seagull. Use the gull1.wav sound file that is a data file for this text.

### Exercise 7

### DigitalClock

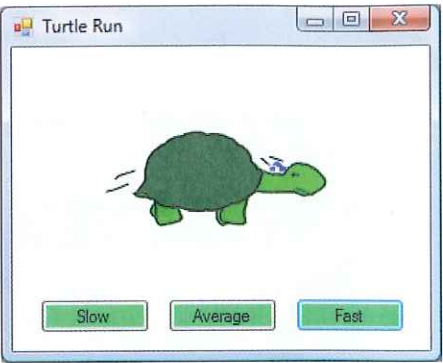
In Chapter 2, Exercise 13, a SystemClock application was created that displayed the current time. Use a timer to create a similar application that displays the time as a running digital clock. *Hint:* Set the time interval to 1000 and create a Timer1\_Tick event procedure that uses the system clock TimeString property.



## Exercise 8

## TurtleRun

Create a TurtleRun application that allows the user to control the speed of the animation by clicking a button. Use the `turtle1.bmp`, `turtle2.bmp`, and `turtle3.bmp` images that are data files for this text. The background color of the form should be white and the background color of the buttons should be MediumSeaGreen. The application interface should look similar to:



## Exercise 9



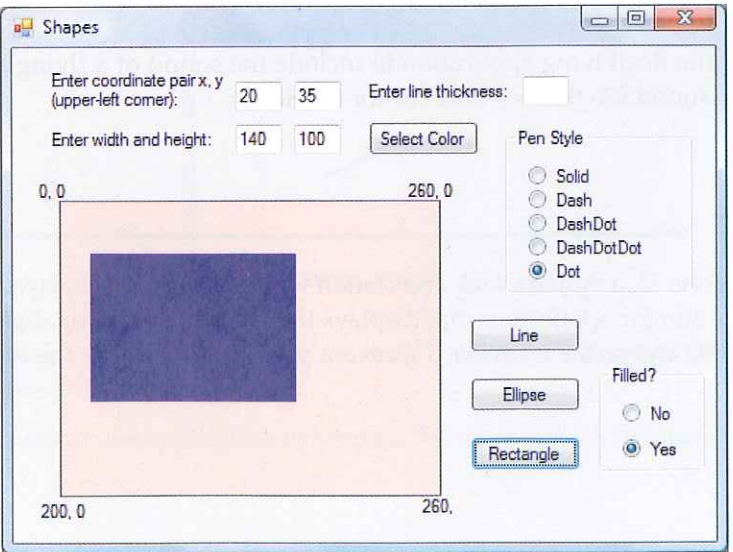
## BouncingBall

Modify the BouncingBall application created in a review in this chapter to include a MouseDown event procedure that reverses the direction of the ball when the form is clicked. For example, if the ball is falling down from the top of the form, clicking the form makes the ball move from the bottom to the top.

## Exercise 10

## Shapes

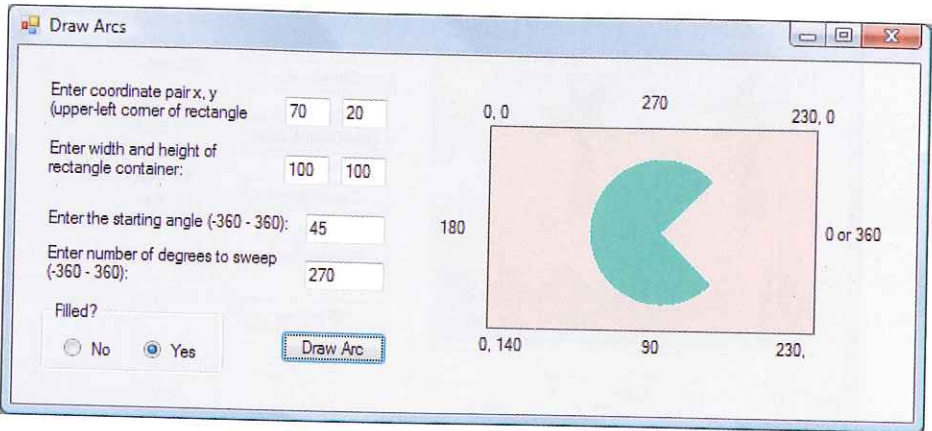
Create a Shapes application that allows the user to enter the information needed to draw a line, ellipse, or rectangle in a MistyRose background colored label. The Select Color button should display a Color dialog box for the user to choose from. Hint: Make the color assignment to a global variable. The application interface should look similar to:



## Exercise 11

## DrawArcs

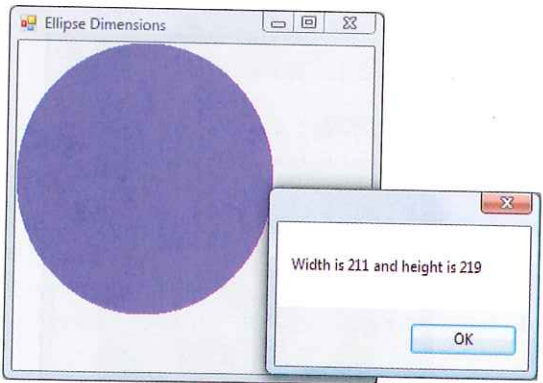
Create a DrawArcs application that allows the user to enter the information needed to draw an arc in a MistyRose background colored label. The application interface should look similar to:



## Exercise 12

## EllipseDimensions

Create an EllipseDimensions application that draws a solid purple ellipse from boundary coordinates (0, 0) in the width and height determined by where the user clicks the form. Indicate in a message box the width and height dimensions of the ellipse. Every time the form is clicked, it should be cleared before drawing the new ellipse and displaying a message box. The application interface should look similar to:

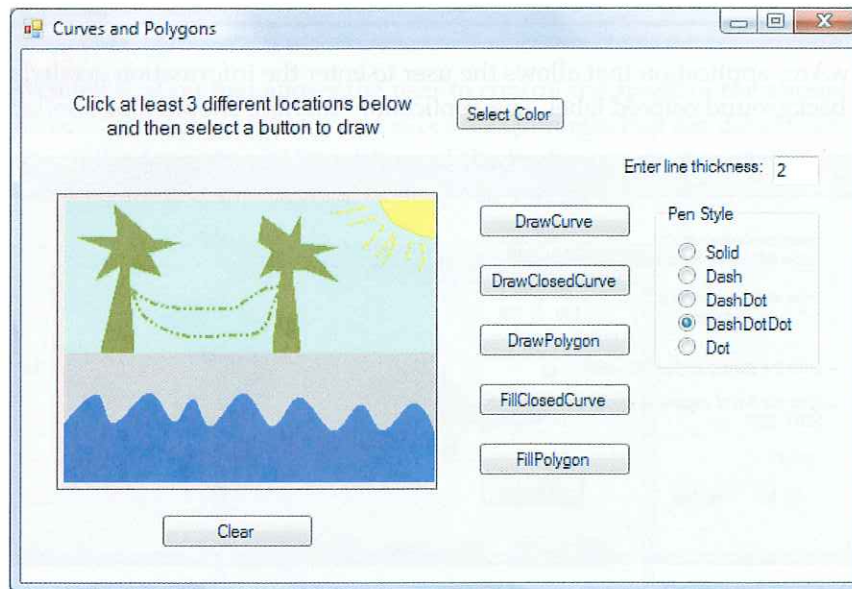


## Exercise 13

## CurvesAndPolygons

Create a CurvesAndPolygons application that allows the user to draw in a label by clicking the mouse to define the points of the curve or polygon before clicking the button indicating the shape to draw. If the user clicks less than three times, display a message box with an appropriate message. The Select Color button should display a Color dialog box for the user to choose from. Hint: Make the color assignment to a global variable. The Clear button should clear the label. The application interface should look similar to the following after drawing many shapes and curves:





## Exercise 14



## Face

Modify the Face application created in a review in this chapter to include a Wink button that, when clicked, winks an eye every second. The Wink button should be disabled until the Happy or Sad button is clicked. Clicking Wink starts the timer and Happy and Sad stops the timer. Your application interface should look similar to the following after clicking Happy and then Wink:



## Exercise 15 (advanced)

## PieChart

Create a PieChart application that allows the user to enter four values to represent relative pie slices in a pie chart. The percentage of the total that each value represents and a pie chart drawing with different color pie slices should be displayed when the Chart button is clicked. *Hint:* Use the percentage of the total values to determine the degree each pie slice will sweep out of the 360 degrees available.