8. What object is created in memory when an exception is thrown and has various properties that contain data about the exception?
9. What is the purpose of a `try-catch` statement?
10. Which class in the .NET Framework provides predefined named constants that are assigned the mathematical values for *pi* and *e*?
11. In code, what function do you call to move the focus to a control?
12. What property allows you to change the color of a control's text?

Programming Problems

1. **Name Formatter**

   Create an application that lets the user enter the following pieces of data:
   - The user's first name
   - The user's middle name
   - The user's last name
   - The user's preferred title (Mr., Mrs., Ms., Dr., etc.)

   Assume the user has entered the following data:
   - First name: *Kelly*
   - Middle name: *Jane*
   - Last name: *Smith*
   - Title: *Ms.*

   The application should have buttons that display the user's name formatted in the following ways:

   Ms. Kelly Jane Smith
   Kelly Jane Smith
   Kelly Smith
   Smith, Kelly Jane, Ms.
   Smith, Kelly Jane
   Smith, Kelly

2. **Tip, Tax, and Total**

   VideoNote
   Solving the Tip, Tax, and Total Problem

   Create an application that lets the user enter the food charge for a meal at a restaurant. When a button is clicked, the application should calculate and display the amount of a 15 percent tip, 7 percent sales tax, and the total of all three amounts.

3. **Distance Traveled**

   Assuming there are no accidents or delays, the distance that a car travels down an interstate highway can be calculated with the following formula:

   $$Distance = Speed \times Time$$

   Create an application that allows the user to enter a car's speed in miles per hour.

   The application should have buttons that display the following:
   - The distance the car will travel in 5 hours
   - The distance the car will travel in 8 hours
   - The distance the car will travel in 12 hours

4. **Sales Tax and Total**

   Create an application that allows the user to enter the amount of a purchase. The program should then calculate the state and county sales tax. Assume the state sales tax is 4 percent and the county sales tax is 2 percent. The program should display

the amount of the purchase, the state sales tax, the county sales tax, the total sales tax, and the total of the sale (which is the sum of the amount of purchase plus the total sales tax).

5. **Celsius and Fahrenheit Temperature Converter**

   Assuming that C is a Celsius temperature, the following formula converts the temperature to Fahrenheit:

$$F = \frac{9}{5}C + 32$$

   Assuming that F is a Fahrenheit temperature, the following formula converts the temperature to Celsius:

$$C = \frac{5}{9}(F - 32)$$

   Create an application that allows the user to enter a temperature. The application should have Button controls described as follows:

   - A button that reads *Convert to Fahrenheit*. If the user clicks this button, the application should treat the temperature that is entered as a Celsius temperature and convert it to Fahrenheit.
   - A button that reads *Convert to Celsius*. If the user clicks this button, the application should treat the temperature that is entered as a Fahrenheit temperature, and convert it to Celsius.

6. **Body Mass Index**

   Create an application that lets the user enter his or her weight (in pounds) and height (in inches). The application should display the user's body mass index (BMI). The BMI is often used to determine whether a person is overweight or underweight for his or her height. A person's BMI is calculated with the following formula:

$$BMI = weight \times 703 \div height^2$$

7. **Sentence Builder**

   The form in Figure 3-41 contains buttons showing various words, phrases, and punctuation. Create an application with a form similar to this one. When the application runs, the user clicks the buttons to build a sentence, which is shown in a Label control. You can use the same buttons as shown in the figure or make up your own. The *Reset* button should clear the sentence so the user can start over.

Figure 3-41 The *Sentence Builder* form