PROGRAMMING ERRORS

There are three types of errors: Errors are usually underlined red, blue or green.

1. syntax errors
2. run-time errors
3. logic errors

SYNTAX ERRORS

- A statement that violates the rules of C# is a syntax error. For example, the second statement is not syntactically correct because constant assignment is illegal outside the declaration.

```
const double PI = 3.14;
PI = 3.141;                    'Syntax error
```

LOGIC ERRORS

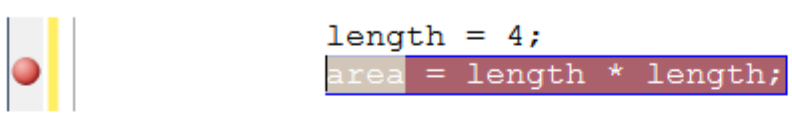
- A logic error is more difficult to detect. Logic errors are caused by statements that are syntactically correct, but produce undesired or unexpected results.

RUN-TIME ERRORS

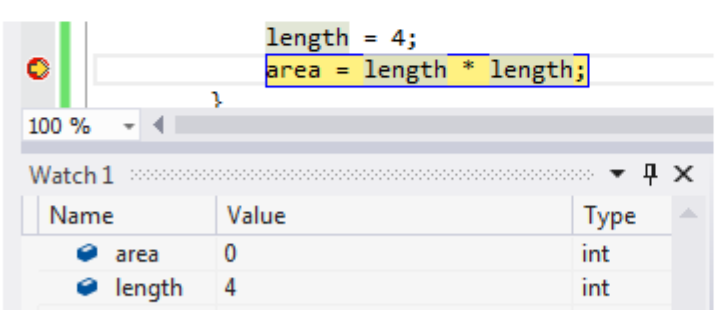
- Errors that are not detected by the compiler may generate a run-time error. A run-time error, also called an exception, halts program execution at the statement that cannot be executed. The statement causing the error is highlighted and an exception helper box is displayed.

DEBUGGING AN APPLICATION

- **Debugging** is the process of getting an application to work correctly. One debugging technique uses breakpoints. A breakpoint is a statement that has been marked as a stopping point. The code below shows a breakpoint, which is highlighted in red:



- A **breakpoint** is created by clicking in the gray area to the left of a statement. When the application is run, program execution stops at the first breakpoint and the IDE goes into break mode. In **break mode**, the **Locals and Watch Windows** can be used to examine values. *Right-clicking* a variable, constant, or object name displays a menu with an Add Watch command. Selecting this command to add the variable, constant, or object name to the Watch window with its current value:



- Program execution is continued from a breakpoint by clicking the **Step Into** Button on the toolbar or pressing the **F8** key, which executes one statement at a time.

- **Debug > Step Into** can also be used to step through a program. Values in the Watch window are automatically updated while stepping through a program.

COMMENTING OUT

- Another debugging technique involves selecting lines of code and then clicking the "Comment out the selected lines" button . Commented statements will not be executed.

- The "Uncomment the selected lines." button on the toolbar removes the double slashes (//) from the beginning of the selected lines of code.

RUNTIME EXCEPTIONS

As an example, attempting to divide by zero in a program will generate an error and C# will throw an exception. If you click on the "Details" button

- You could click "continue", but a program typically will not run very well after an exception is thrown.

- If you click "quit", the program will end immediately.

- We will explore how to avoid and what you can do about runtime exceptions later on.