100 XP

# Introduction

3 minutes

Selection and iteration statements use code blocks to group-together the code lines that should be executed, skipped, or iterated over. But that's not the only purpose for code blocks. Code blocks can also be used to control or limit variable accessibility. Variable "scope" refers to the portion of an application where a variable is accessible. Understanding how a code block affects variable scope is an important part of computer programming.

Suppose you're working on large application that uses nested iteration and selection statements to process array data. Your application uses variables to help accomplish common tasks throughout the application. Some variables serve the same purpose in different portions of the application, and you've made some attempt to reuse the variable names. As your application grows, you start seeing unexpected results for calculations, and errors that report a variable that is uninitialized or doesn't exist. You need to improve the approach you're using to declare and access variables, and you need to improve your understanding of variable scope.

In this module, you'll declare variables for use inside and outside the boundaries of code blocks. You'll remove code blocks in certain situations to make code more readable. You'll learn how code blocks affect the accessibility and visibility of your variables.

By the end of this module, you'll be able to use code blocks with more confidence, understanding how they impact the visibility and accessibility of your code.

## Learning objectives

In this module, you will:

- Understand the impact of declaring and initializing variables inside and outside of code blocks.
- Improve the readability code blocks in `if` statements.

## Prerequisites:

- Experience declaring and initializing variables.

- Experience with `if-elseif-else` selection statement structures.
- Experience with `foreach` iteration statements.
- Experience calling methods of classes in the .NET Class Library.

## Next unit: Exercise - Code blocks and variable scope

Continue >