100 XP

Introduction

3 minutes

The C# programming language is similar to any human written or spoken language. They each support different ways of expressing the same idea. In spoken languages, some words and phrases are more descriptive, accurate, or succinct than others. In the C# programming language, there is more than one way to create branching logic. For example, selections that use `if` statements and selections that use `switch` statements. Depending on the context of your application, one type of selection statement might be more expressive and succinct than the other.

Suppose working on applications that make extensive use of selections statements. In some cases, `if-elseif-else` constructs are used to produce succinct and expressive code that is easy to read and maintain. In other cases, the `if-elseif-else` constructs produce the required result, but are difficult to read and maintain. You have been tasked with reviewing the code and determining when it is suitable to use a `switch` statement rather than an `if` statement.

In this module, you'll investigate the use of a `switch` statement to implement branching logic as an alternative to an `if` statement. You'll also work on converting an `if-elseif-else` construct to a `switch-case` construct. During this process, you'll learn to recognize the benefits of choosing one type of selection statement over the other.

By the end of this module, you'll be able to implement `switch` statements in your application, judge when to use a `switch` statement over an `if-elseif-else` construct, and convert `if-elseif-else` constructs to `switch` statements.

Learning objectives

In this module, you will:

- Use the `switch-case` construct to match a variable or expression against several possible outcomes.
- Convert code that uses an `if-elseif-else` construct into a `switch-case` construct.

Prerequisites:

- Experience using the `if-elseif-else` construct for adding branching logic.
- Experience working with variables, string interpolation, and printing output.

Next unit: Exercise - Implement a switch statement

Continue >