

Ref.:

<https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/statements/selection-statements>

<https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/statements/iteration-statements>

<https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/constants>

the *do-while* loop

Use it when ...

- you want to execute a set of instructions at least ONCE before checking a condition
- you want to validate user's input
- you want to repeat the play

```
do
{
    Console.WriteLine(@"This is the body of the do-while loop.
        Control of the program arrives to the do-while loop and walks into the body.
        It begins executing the instructions in the body.
        When it's done, it moves to the while and evaluates the condition.
        If true, control of the program goes back to the beginning of the do;
        If false, control of the program continues on past the do-while.
        Each full pass through the loop (top to bottom) is called an iteration.");
} while (true); // if the cdt is always true, the loop becomes infinite
// } while (false) ; // if the cdt is always false, the loop will always execute only once
```

```
byte num;
bool validInput;

do
{
    Console.WriteLine("Enter a number between 1 and 7: "); // prompt with a constraint
    num = Convert.ToByte(Console.ReadLine()); // read it in, assuming it's the right type

    validInput = num >= 1 && num <= 7; // check if it's valid

} while (validInput == false); // set the cdt accordingly
//} while (!(num >= 1 && num <= 7)) ; // !validInput --> invalidInput
//} while (num < 1 || num > 7) ; // when you're comfortable, skip the boolean and capture the cdt
```

```
char ans;

do
{
    // Let's play a game!
    // Game code goes here...
    // Gameplay done with "Congratulations, you won!" or "Sorry, you lost!"

    // prompt to see if user would like to go again

    Console.WriteLine("Would you like to play again? (y/n)");
    ans = Convert.ToChar(Console.ReadLine());

} while (ans == 'y');
//} while (ans.Equals('y'));
```

#1 [nested switch] Prompt the user for two numbers (int or float) and an operator.

- a. Use a switch statement to execute the appropriate instructions assuming the operator entered is an arithmetic operator (+, -, *, / or %). Read the arithmetic operator as a char.
- b. Use another switch statement to execute the appropriate instructions assuming the operator entered is a relational operator (<, <=, >=, >, ==, !=). Read the relational operator as a string.
- c. Nest the switch statements you developed in 1a & 1b into an outer switch. Ask the user if they would like to perform an arithmetic operation (choose '1' or 'a') or a relational operation (choose '2' or 'r'). Use the appropriate datatype to store their answer and direct them to the relevant case.

The switch statement for the above will have the following syntax:

```
switch (/*var*/)
{
    case /*value*/:
        switch (/*var*/)
        {
            case /*value*/:
                // instructions here
                break;
            // other cases here
            // potentially default case
        }
        break;
    case /*value*/:
        switch (/*var*/)
        {
            case /*value*/:
                // instructions here
                break;
            // other cases here
            // potentially default case
        }
        break;
}
```

// switch – coming soon

#2 [do-while & switch]

- Prompt the user for a temperature between -50 and +5 °C. Embed the prompt in a do-while to validate the input; the condition in the while should be satisfied when the input is out of range.
- Now, prompt the user for a wind speed between 5 and 80 km/h. Embed this prompt in another do-while loop to validate the input; again, the condition in the while should be satisfied when the input is out of range.
- Use the weather chart below from [santemontreal](http://santemontreal.com) to let the user know whether it's risky, comfortable, or dangerous for the children to play outside.

Wind Chill Chart⁵

Actual Air Temperature (°C)

	5	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50
5	4	-2	-7	-13	-19	-24	-30	-36	-41	-47	-53	-58
10	3	-3	-9	-15	-21	-27	-33	-39	-45	-51	-57	-63
15	2	-4	-11	-17	-23	-29	-35	-41	-48	-54	-60	-66
20	1	-5	-12	-18	-24	-30	-37	-43	-49	-56	-62	-68
25	1	-6	-12	-19	-25	-32	-38	-44	-51	-57	-64	-70
30	0	-6	-13	-20	-26	-33	-39	-46	-52	-59	-65	-72
35	0	-7	-14	-20	-27	-33	-40	-47	-53	-60	-66	-73
40	-1	-7	-14	-21	-27	-34	-41	-48	-54	-61	-68	-74
45	-1	-8	-15	-21	-28	-35	-42	-48	-55	-62	-69	-75
50	-1	-8	-15	-22	-29	-35	-42	-49	-56	-63	-69	-76
55	-2	-8	-15	-22	-29	-36	-43	-50	-57	-63	-70	-77
60	-2	-9	-16	-23	-30	-36	-43	-50	-57	-64	-71	-78
65	-2	-9	-16	-23	-30	-37	-44	-51	-58	-65	-72	-79
70	-2	-9	-16	-23	-30	-37	-44	-51	-58	-65	-72	-80
75	-3	-10	-17	-24	-31	-38	-45	-52	-59	-66	-73	-80
80	-3	-10	-17	-24	-31	-38	-45	-52	-60	-67	-74	-81

Wind Speed (Km/h)

What to do when the temperature is in the yellow zone:

- ❄ Make sure the children are well dressed before going outside.
- ❄ Encourage the children to be physically active.
- ❄ Stay outside for a shorter period.
- ❄ Be on the lookout because some children may be cold.

Please print and put up the wind chill chart.

- Comfortable
- Risky
- Dangerous

3

#3 [input validation on grades]

- Prompt the user for a grade between 0 and 100. Use a do-while to validate the input. The condition needs to be satisfied for invalid input.
- Prompt the user for a letter grade that falls between 'A' and 'F', except 'E'. Use a do-while to validate the input. *Hint: there will be 3 parts to this condition, pay attention to the precedence of the logical operators.*

#4 [dual input validation] Prompt the user for a month m and a day d at the same time. Then, embed your prompt in a do-while loop to validate the input.

#5 [from Assignment 3]

- Modify ex. 3 to make sure no negative number of days is entered.
- Modify ex. 6 to make sure the year is 4 digits.
- Modify ex. 7 and 8 to make sure the $1 \leq m \leq 12$, $1 \leq d \leq 31$, and y is 4 digits.

#6 [constants & pin] Think of constants as variables whose value *should not* and *do not* “vary”. You can use them, but not change their values. To create a constant in C#, use the *const* followed by the variable name in Pascal case notation.

```
const float PassingGrade = 60.0; // a variable prefixed with the keyword const
```

Define or declare a constant *my_pin* and give it a 4 or more-digit value. Then, prompt the user for the pin. Embed the prompt in a do-while. The condition must satisfy an incorrect pin to re-prompt the user.

#7 [repeat the play]

- a. Embed ex 6, 7 or 8, from Assignment 3, in a do-while() loop. At the end of the do-while, before you reach the closing bracket {}), ask the user: “Would you like to enter another date? (y/n)”. Read the answer in a *char ans*. What should the condition of the while satisfy to repeat the play?
- b. What is your observation? How does this do-while work?

#8 [nested do-while – validation & repetition] In the exercises above, you have seen how a do-while can be used for input validation and to repeat a play. In fact, you can use a do-while to repeat any given set of instructions. Imagine these instructions are meant to read-in a set of grades. So, you want to build a do-while in which you prompt the user to enter a grade. Then you want to ask them if they want to enter another grade. If the answer is yes, the condition of the while will be satisfied, and you will walk into the do body again.

- a. Embed the grade-prompt in a do-while that validates the grade.
- b. Embed this do-while in an outer do-while that asks the user if they want to enter another grade.
- c. Once you’ve validated the grade, and before prompting the user, add the grade to a variable sum
- d. Once you exit the outer do-while, compute the average. Did you count how many grades have been entered?

#9 [switch – menu] Print the following (feel free to make it your own!):

```
***** Welcome to Myflix! *****
```

What are you in the mood for, <insert_name_here>?

- 1 – Non-stop Awesome Action
- 2 – Superhero & Sci-Fi
- 3 – Romantic but Lit-wise Deep
- 4 – Fun & Uplifting Adventures
- 5 – Kids Time
- 6 – Educational
- 7 - Your Faves

- a. Read in the user’s choice (number between 1 – 7) and use a do-while to validate it.
- b. Use a switch statement to print the available options in each category or pretend to play something randomly.
- c. Ask the user if they would like to watch another movie (y/n). Use a do-while to validate their answer.
- d. Embed your switch in a do-while that would handle repetition.