Contents

do…while Loops

This lab serves multiple goals:

- To reinforce your understanding of the syntax of `do…while` loops,
- To stress the differences between **`while`** and `do…while` loops, and
- (Optional) To use a `do…while` loop to solve a simple problem.

Implementing do…while Loops – Warm-up

For all the problems in this section, use a `do…while` loop.

1. Write a program that displays the numbers 0 to 50.
2. Write a program that displays the numbers 30 to -20.
3. Write a `do…while` loop that generates this output:

```
1 10 100 1000 10000 100000 1000000
```

User Input Validation using do…while Loops

In the following problem, implement a program combining a `do…while` loop with user input that satisfies the following requirements:

1. Ask the user to enter an integer between (and including) 0 and 100.
2. If the value provided by the user is not in this range, the program should repeat the question.
3. After the user provides an integer within the range, display that number.

Here is an example of execution, where the user input is <u>underlined,</u> and hitting “enter” is represented by “↵”:

```
Enter an integer between 0 and 100:
NO↵
Sorry, that is not a valid integer.

Enter an integer between 0 and 100:
-20↵
```

```
Sorry, that is not a valid integer.

Enter an integer between 0 and 100:
42↵
You entered 42.
```

Comparing `while` and do…`while` loops

Consider the program given below that has been implemented using a **`while`** loop:

```csharp
int n;
bool flag = false;

Console.Write("Enter an integer:");
flag = int.TryParse(Console.ReadLine(), out n);

while(!flag)
{
    Console.WriteLine("The value you entered is not a valid integer.");
    Console.Write("Enter an integer:");
    flag = int.TryParse(Console.ReadLine(), out n);
}

Console.WriteLine($"The number you entered is {n}");
```

1. Convert the program to an equivalent version that uses a do…`while` loop.
2. Test the do…`while` version with different inputs to ensure the behavior is the same and that the program does not crash with an error. For example, you should try:
   - alphabetic input (e.g., "Train") which would be invalid,
   - floating point numbers (e.g., "12.5") which would be invalid,
   - negative integers (e.g., "-12") which would be valid,
   - positive integers (e.g., "10") which would be valid, and
   - the number 0 which would be valid.
3. Compare the **`while`** and do…`while` implementations: which one is better, in your opinion, and why?

Pushing Further (Optional)

This time, you are given a simple task without much guidance. This is very similar to the type of questions you may face during exams:

> Write a program that asks the user to enter a temperature in degrees Fahrenheit. If the user enters something that is not a

number or enters a number outside the “sensible” range of -20 to 120 degrees (both included), your program should ask for the temperature again, and it should keep asking until the user provides valid input.

Once the user has entered a “sensible” temperature value, your program should initialize a string variable named description to one of the following values: “Cold” if the temperature is below 40 degrees, “Mild” if the temperature is between 40 and 70 degrees, or “Warm” if the temperature is above 70 degrees. This string should then be displayed.

A possible solution is given below.

Solution:

A possible solution, using `do…while` is:

```
int temp = 0;
bool tempConvert;
do{
    Console.Write("What's the current temperature outside?: ");
    string strTemp = Console.ReadLine();
    tempConvert = int.TryParse(strTemp, out temp);
    if (tempConvert == false)
        Console.WriteLine("That's not a temperature, that's a word.");
    else if (temp <= -20 || temp >= 120)
        Console.WriteLine("That's not possible. Be serious.");
} while (temp <= -20 || temp >= 120 || tempConvert == false);

string description;

if (temp < 40) {description = "cold";}
else if (temp <= 70){description = "mild";} // Note that we know that temp >= 40 holds at this point.
else {description = "warm";} // Note that we know that temp > 70 holds at this point.

Console.WriteLine($"Wow, it's {description} outside today...");
```