

# Class With Decisions

<https://csci-1301.github.io/about#authors>

June 12, 2021 (11:27:36 AM)

## Contents

### 1 Validating Inputs

1

## 1 Validating Inputs

Start by downloading the `LoanCalculator`<sup>1</sup> solution, that mixes classes and decision structures.

Edit the `Program.cs` file of the project you previously downloaded to add the following validation features:

1. Users entering a value other than `A`, `a`, `H`, `h`, `O` and `o` for the loan type will be asked again, and asked as long as they do not give a valid answer.
2. Users entering a credit score not between 300 and 850, or that is not an integer, will be asked again, and asked as long as they do not give a valid answer.
3. Users entering an amount needed or a down payment that is not a decimal, or is a negative decimal, will be asked again, and asked as long as they do not give a valid answer.
4. (Optional) Use the `ToLower()`<sup>2</sup> or `ToUpper()`<sup>3</sup> methods of the `char` class to make the program more readable – you will be able to greatly simplify the `if` statement that checks the loan type.
5. (Optional, hard) Write a method for the `Loan` class that takes a character as an argument, and returns the string describing the type of loan designed by that character. Then, use this method in the `ToString` and in the application program instead of doing it “by hand”.

You can find a possible solution in this archive<sup>4</sup>.

---

<sup>1</sup>`LoanCalculator.zip`

<sup>2</sup><https://docs.microsoft.com/en-us/dotnet/api/system.char.ToLower?view=netframework-4.7.2>

<sup>3</sup><https://docs.microsoft.com/en-us/dotnet/api/system.char.ToUpper?view=netframework-4.7.2>

<sup>4</sup>`LoanCalculatorSol.zip`