NUMERIC TYPES SUGGESTER

Ultimate C# Masterclass Assignment

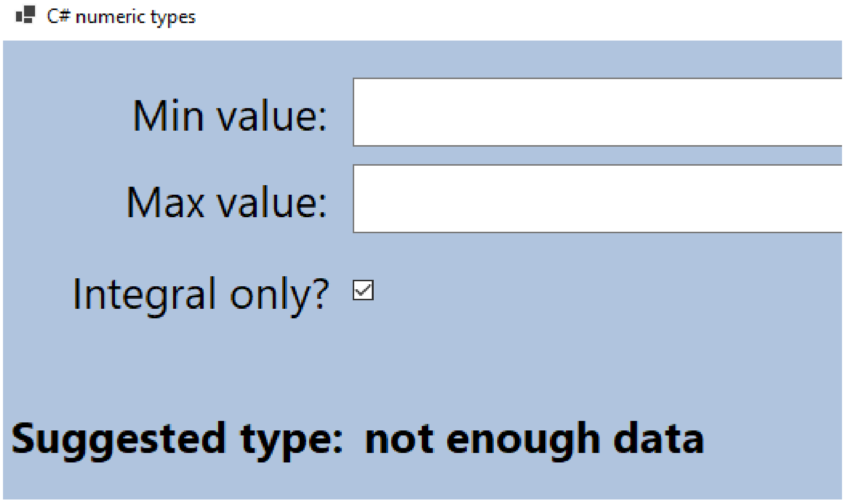
Overview

This app's purpose is to help C# developers choose the best numeric types for their needs.	Windows Forms App

This app is meant to help C# developers choose a **proper numeric type for their needs**. The user enters the **minimal** and **maximal** value the type should support and also checks checkboxes if the numbers **should only be integral**, and if not, then if as floating-point numbers, they must be represented **precisely**.

As a result, the app shows the smallest numeric type fit for the user's needs. For example, if the user selects 0 as the minimum value and 100 as the maximum value and also decides the numbers should only be integral, then the app will suggest choosing **byte**.

The initial view of the form:



C# numeric types

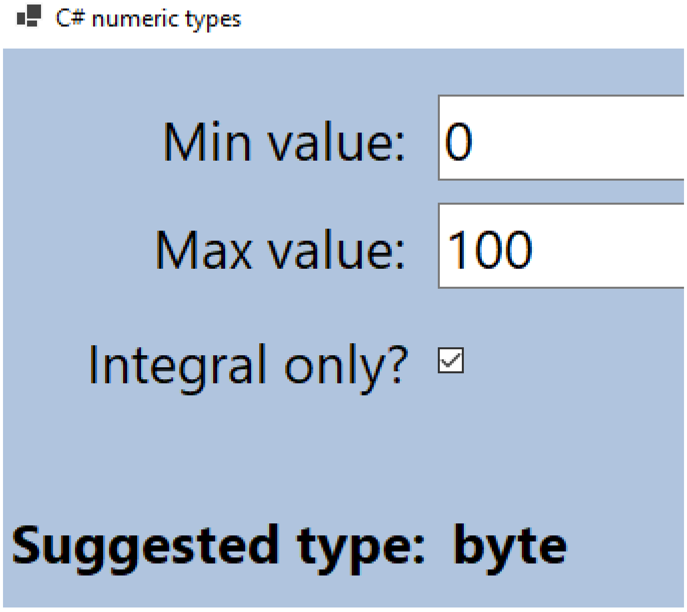
Min value:

Max value:

Integral only? ☑

Suggested type: not enough data

The view of the form after valid values are entered:



C# numeric types

Min value: 0

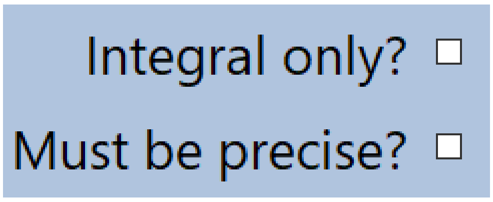
Max value: 100

Integral only? ☑

Suggested type: byte

Elements of the form:

- **Min value**: A text field in which the user can enter any integral number. The only non-digit character that can be entered is the minus sign, which can be added as the first character to indicate the number is negative. Initially it should be empty. Since this field can store enormous numbers, its value should be parsed to **BigInteger**.
- **Max value**: Should behave the same as the Min Value field. This field is **validated**. See "**Validation**" for more information.
- **Integral only?**: A checkbox indicating if only integral numbers should be supported. Initially should be checked.
- **Must be precise?**: A checkbox indicating if floating-point numbers should be precisely represented. It is **only shown** if the "Integral only?" checkbox is **unchecked**. Initially should be unchecked.



Integral only? □

Must be precise? □

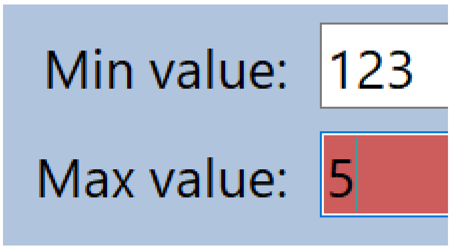
- **Suggested type**: A label showing the suggested numeric type. Initially set to "Not enough data".

Validation:

The state of the form is valid if two conditions are met:

- Both **Min value** and **Max value** fields are not empty and represent valid numbers (if the only character entered is the minus sign, this means it is not a valid number)
- **Max value** is no smaller than **Min value**

If both **Min value** and **Max value** are non-empty, and the Max value is smaller than Min value, the Max value field should be colored in red:



Min value: 123

Max value: 5

Triggering recalculation of suggested type:

The **Suggested type** should be recalculated each time the state of the form is changed (the Min or Max values are changed, or any checkbox is checked or unchecked). If, after the change, the state of the form is invalid (text boxes are empty or Max Value is smaller than Min Value), then "not enough data" should be printed in place of the suggested type.

Calculation of suggested type:

The smallest type that can store the numbers meeting the given criteria should be used. For example, if the minimum value is 0 and the maximum is 30 000, then the suggested type should be **ushort**, not int.

If the minimum value is zero or more, the unsigned types should be chosen if possible.

It is important to take not only the maximal value into consideration but also the minimal. For example, if the max value is 0, but the min value is minus 4 billion, then the suggested type must be long.

Not all numbers are possible to be represented. There are two cases when there is no type that can store the numbers specified in the form:

- When the **Integral only** is unchecked, **Must be precise** is checked, but the values are beyond the range of the **decimal** type.
- When the **Integral only** is unchecked, **Must be precise** is unchecked, but the values are beyond the range of the **double** type.

In both those cases, "Impossible representation" should be printed in place of the suggested type.

Examples of suggested types:

Min value	Max value	Integral only?	Must be precise?	Suggested type
0	1 000 000	🟩		uint
-1	1 000 000	🟩		int
0	larger than ulong.MaxValue	🟩		BigInteger
smaller than long.MinValue	1	🟩		BigInteger
-30 000	30 000	🟩		short
-1	5	🟩		sbyte
0	100 000	🟥	🟥	float
-1	100	🟥	🟩	decimal
smaller than float.MinValue	0	🟥	🟥	double
0	larger than decimal.MaxValue	🟥	🟩	Impossible representation
smaller than double.MinValue		🟥	🟥	Impossible representation