

Variables, Expressions, and Statements

Module 1

Quick Review

- Getting around Visual Studio
 - Projects
 - Files
 - Windows
- WPF programming (Windows Presentation Foundation)
 - Controls
 - Properties
 - Events



Variables

- How programs keep track of data
- Variables have four things:
 - Identifier (name)
 - Type (string, int, etc.)
 - Value
 - Scope (lifespan of the variable) [***To be discussed later***]



Rules for Variable Names - Syntax

- Naming Syntax (will cause errors if you don't follow)
 - Enforced by the compiler
 - Identifiers can contain letters, numbers, and underscore (_)
 - Can't start with a number or contain spaces or special characters
 - Case matters! (**MyVariable** is not the same as **myvariable**)



Rules for Variable Names - Convention

- Naming Convention (will cause confusion if you don't follow)
 - Enforced by Mr. Brooks 🐱
 - Name should indicate what the data is for (meaningful)
 - “camelCase”: start with lowercase letter, then each subsequent word starts with a capital letter (C#)
 - Names shouldn't contain non-standard abbreviations

Type

- Numbers - `int`
 - Whole numbers only
- Letters - `string`
 - `null`, `""`, "Or any string of characters"
- And many others we won't discuss today



Expressions

- Something that evaluates to a single **value**
- Has a **type**
- Use **operators** to combine expressions



Arithmetic Operators (numeric values only)

<u>Operator</u>	<u>Example</u>	<u>Answer</u>
+ (addition)	10 + 3	
- (subtraction)	10 - 3	
* (multiplication)	10 * 3	
/ (division)	10 / 3	
% (modulus or remainder)	10 % 3	



String Operators (string and numeric values)

<u>Operator</u>	<u>Example</u>	<u>Answer</u>
+ (concatenation)	“Cut” + “e”	
+ (concatenation)	“10” + “3”	
+ (concatenation)	“10” + 3	
+ (concatenation)	10 + “3”	
+ (concatenation)	10 + 3	



Operator Precedence

- What is the order of operation?
 - $4 + 3 * 2 / 3 - 1$
- What is the value?
 - 3, 5, or 7? Or something else?
- Rule:
 - First priority: $*$ / $\%$
 - Second priority: $+$ -
 - When same priority, left - right



Statements

- A single, complete instruction for the program
- In C#, all statements end with a statement separator (;)
- In C#, use braces ({ and }) to specify a block of statements
- All statements in a block are executed in order



Worksheet & Assignment

