C# Fundamentals 1 (CSF1)

Jeremy Rutherford

Course Objectives

- Describe the key parts of the .NET architecture.
- Create a simple .NET console application using Visual Studio 2015 Community.
- Define the fundamental data types used in .NET applications.
- Demonstrate the ability to declare and assign a variable using C#.
- Understand commonly-used naming conventions.
- Identify and explain how to use operators in a .NET application.
- Determine the appropriate control structure to use in a given scenario.
- Understand the benefits of using an Object-Oriented Programming language.
- Use string formatting to customize the output of data into a console window.
- Understand and use basic collections to store multiple values.
- Perform basic debugging in a .Net application

Module 1:
Introduction to C# and .NET

Intro to C# - Objectives

- Discuss the .NET Framework
- Understand the key traits of C#
- Explain the compilation process
- Identify the three basic types of errors

.NET Overview

- Mainly runs on Windows
- Like a mini-OS or gaming console
- Multi-Language support via a Common Type System (CTS)

- Less interaction with the System Registry
- Easier Versioning
- Manages Code Execution

- Framework Class Library (FCL)
- Built in Security
- Object Oriented (OO)

C# Traits

- Specifically written for the .NET Framework
- Object Oriented Programming (OOP)

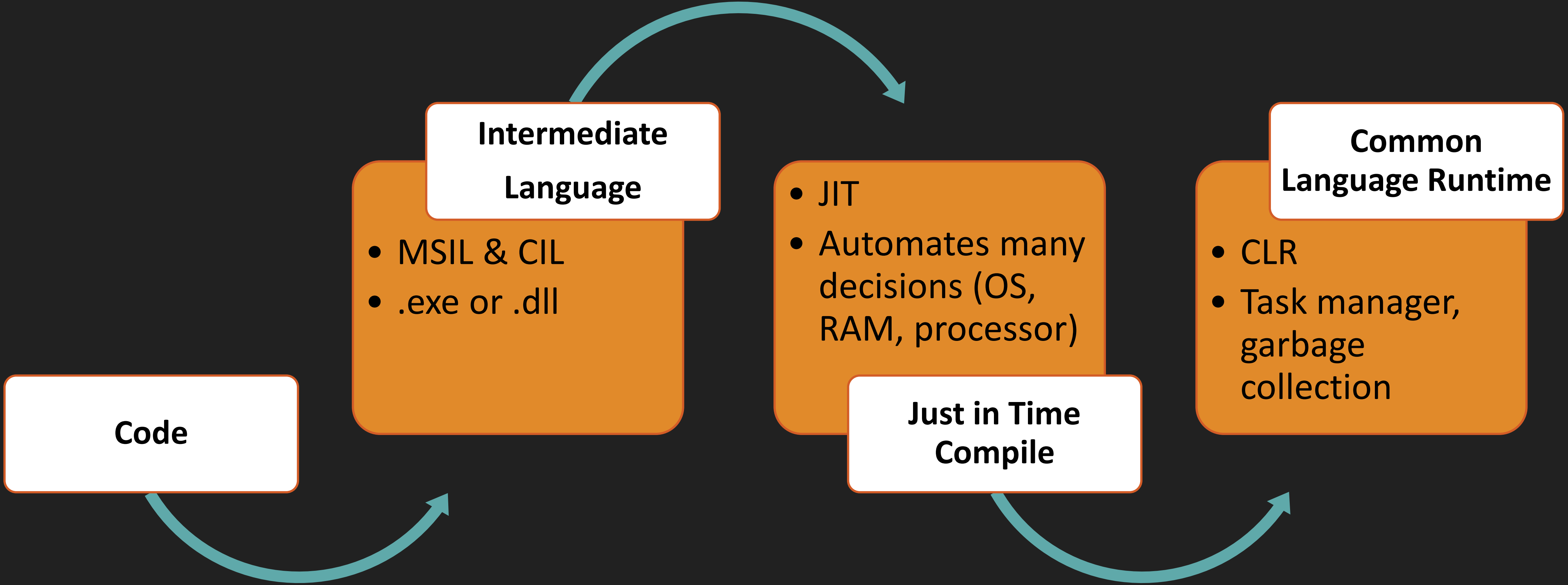
- Designed to be managed
- Similar to Java and C++

- Case Sensitive
- Uses curly braces {}

- Semi-colons are like periods
- Largely ignores whitespace

- Type Safe
- File extension .cs

Compilation Process



Intro to C# - Errors

- Syntax:
  - the code was written wrong
- Runtime:
  - Syntax is correct, but code encounters an unexpected error during execution
- Logic:
  - The logic written yields unexpected results

END MODULE 1

Homework:

1. Quizlet Vocabulary

- .NET Framework
- Key traits of C#
- Compilation process
- Identify the three basic types of errors

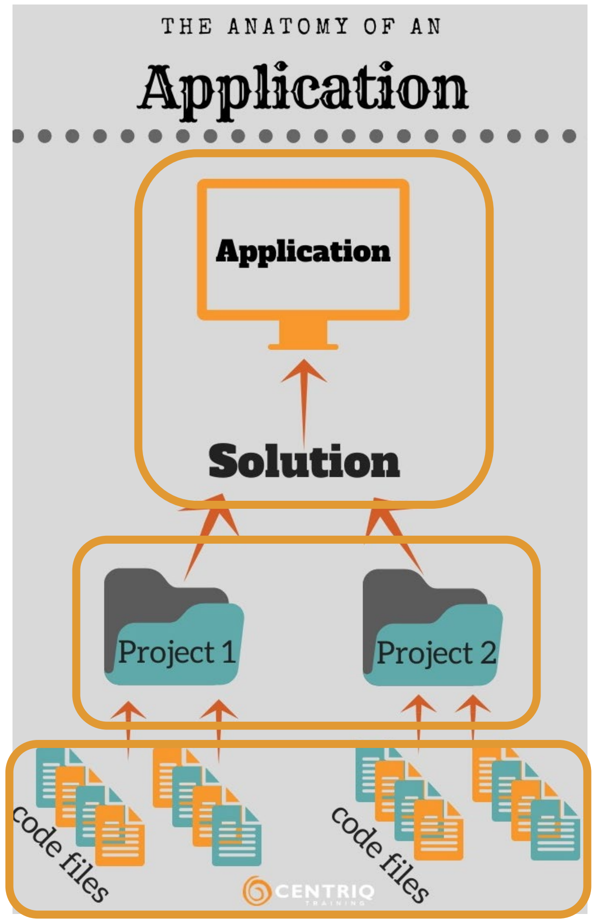
Module 2: Introduction to Variables

Intro to Variables – Objectives

- Understand the role of variables in Code.
- Utilize two data types in C# to create variables
- Demonstrate how to make code comments.
- Understand the basic rules of the C# language & how it’s written.

Intro to Variables – Anatomy of an App

- Solution (.sln) – The application. Contains all the basic files necessary to run a program.
- Project: Division within a solution that contains 1 or more code files. a solution will ALWAYS contain at least one project.
- Code Files (CLASS): files that contain code that may be used by your application. A project will ALWAYS contain at least one code file.



CODE ALONG!

END MODULE 2

Homework:

1. Quizlet Vocabulary
2. Read Chapters 1 & 2 in Course Text

- Understand the role of variables in Code.
- Utilize two data types in C# to create variables
- Demonstrate how to make code comments.
- Understand the basic rules of the C# language & how it’s written.

Module 3:
More Data Types

More Data Types – Objectives

- Discover additional intrinsic data types in C#
- Understand how values are stored

More Data Types –
Bob Sure Is Loving!

	Data Type	Lesser Used Variant	Bit (binary digit) size	Value Range
Bob	byte		8	0 to 255
		sbyte(signed)	8	-128 to 127
Sure	short		16	~-32k to ~32k
		ushort(unsigned)	16	0 to ~65k
Is	int		32	~-2bil to ~2bil
		uint	32	0 to ~4bil
Loving	long		64	~ -9quint to ~9 quint
		ulong	64	0 to ~18quint

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0

255

More Data Types –
Bitmap

128	64	32	16	8	4	2	1	Total
1	1	1	1	1	1	1	1	255
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	33

More Data Types – Summary

- String
- Int
- Byte/SByte
- Short/Ushort
- Int/UInt
- Long/Ulong
- Bool
- Char

CODE ALONG!

QUIZ!

Module 1-3 Quiz in Canvas

END MODULE 3

Homework:

1. Quizlet Vocabulary
2. Read Chapters 1 & 2 in Course Text

- 14 Total Datatypes
- Understand how values are stored

Module 4: Naming Conventions

Naming Conventions – Objectives

- Understand naming conventions given to variables.
- Demonstrate the typing convention associated with each naming conventions.

CODE ALONG!

Naming Conventions – Summary

- **UPPERCASE –** all capital letters
- **lowercase –** all lowercase letters
- **Pascal Case –** A capital letter for each word in the name
- **camelCase –** lowercase first letter for the first word, followed by a capital first letter for each following word
- **Hungarian / Lezenski –** camelCase, but the first word represents a description of the type of object the variable is

STOP

END MODULE 4

- Understand naming conventions given to variables
- Demonstrate the typing convention associated with each naming convention

Homework:

1. Quizlet Vocabulary
2. Read chapter 3 in the course text

Module 5: Casting

Casting –
Objectives

- Reflect on How Variables Act as Boxes to Store Information for Later Use.
- Reassign New Values to Change Contents.
- Reassign Values That Come from Other Variables.
- Take a Value of One Data Type and Put Into Variable of Another Data Type.

Casting –
Key Terms

- **Casting –** Copying the value of a variable of one datatype to a variable of a similar, but different datatype
- **Implicit Casting –** copying the value from a smaller datatype to a larger datatype
- **Explicit Casting –** copying the value from a larger datatype to a smaller datatype
- **Truncation –** occurs when the value contained in one datatype cannot fit into a variable of a smaller datatype. This results in lost information as a result of binary bit cells being chopped off permanently.

Casting –
Data Types Chart

	Data Type	Bit size	Value Range	Cast Type	
Bob	byte / sbyte	8	0 to 255 /-128 to 127	↓	↑
Sure	short / ushort	16	~-32k to ~32k / 0 to ~65k	↓	↑
Is	int / uint	32	~-2bil to ~2bil / 0 to ~4bil	↓	↑
Loving	long / ulong	64	~ -9quint to ~9 quint / 0 to ~18quint	↓	↑
				Implicit	Explicit

CODE ALONG!

LAB!

LAB 1: Datatypes Lab

LAB!

LAB 2: Casting Lab

Casting –
Summary

- **Right-to-left thinking –** helpful when building and troubleshooting applications
- **Truncation and casting** – Understanding these is useful in real-world applications in a scenario where you may be losing value unexpectedly from a variable
- **Count and Create a bitmap –** Useful in technical interviews and raises understanding about how technology works behind the scenes

END MODULE 5

END MODULE 5

Homework:

1. Quizlet Vocabulary
2. Complete unfinished labs

- Reflect on How Variables Act as Boxes to Store Information for Later Use.
- Reassign New Values to Change Contents.
- Reassign Values That Come from Other Variables.
- Take a Value of One Data Type and Put Into Variable of Another Data Type.

Module 6: Mathematical Operations

Mathematical Operations-
Objectives

- Perform Mathematical Operations in C#.
- Learn Correct Order of Operations.
- Understand How/When to Use Mathematical Operators in the Language.

Mathematical Operations- Operators

- +, -, *, /
  - Basic addition, subtraction, multiplication, and division
- %
  - Modulus – performs division and returns only the remainder
- Assignment Operators (+=, -=, *=, /=, %=)
  - performs an arithmetic operation and automatically stores the new value in the existing variable
- Unary Operator (++,--)
  - adds or subtracts 1 from the existing variable. Can be pre- or post-fixed.

CODE ALONG!

END MODULE 6

Homework:

1. Quizlet Vocabulary
2. Read Chapter 3
3. Continue typing.io practice (30 minutes)
4. Post something interesting from chapters 1-3 in Canvas

- Perform Mathematical Operations in C#.
- Learn Correct Order of Operations.
- Understand How/When to Use Mathematical Operators in the Language.

Module 7:
Input, Parse, Convert

Input Parse Convert –
Objectives

- Learn how to receive and perform activity with input from a user
- Provide a response to user based upon captured information

Input Parse Convert –
Key Terms

- **Input** – Information captured from the application’s user.
- **Output –** Information shown to the application’s user.
- **String Formatting –** A way to show output to an application’s user by putting information into placeholders within the output string. Allows formatting to be done to the information in the placeholder.
- **Parse / Convert** – Two methods of changing the data stored in a variable of one datatype into data of another non-similar datatype (i.e. string data to int).

CODE ALONG!

LAB!

Water Lab
Change Lab

QUIZ!

Module 4-7 Quiz in Canvas

END MODULE 7

Homework:

1. Quizlet Vocabulary
2. Complete any unfinished labs

- Learn how to receive and perform activity with input from a user
- Provide a response to user based upon captured information

Module 8:
Logical and Comparison Operators

Logic and Comparison – Objectives

- Learn About Different Operators (Outside of Mathematical Operators).
- Review and Use Comparison and Logical Operators.

Logic and Comparison –
Key Terms

- **Comparison Operator –** A character that checks the value of objects on either side of the operator and returns a bool value of true or false.
- **Logical Operator -** A character set that compares the bool values on either side of it and returns a bool value of true or false

CODE ALONG!

Logic and Comparison – Summary

- **Comparison Operators:**
  - > is greater than
  - < is less than
  - >= is greater than or equal to
  - <= is less than or equal to
  - == is equal to
  - != is not equal to
- **Logical Operators:**
  - Combine two comparison operators or bool values and returns a bool
  - && is used for AND
  - | | is used for OR
  - BONUS: ^ is used for EXCLUSIVE OR (XOR) – exactly one argument is true. Returns false if both are true, or both are false.

END MODULE 8

Homework:

1. Quizlet Vocabulary

- Learn About Different Operators (Outside of Mathematical Operators).
- Review and Use Comparison and Logical Operators.

Module 9:
String Formatting

String Formatting-
Objectives

- Understand How to Format String Values.
- Demonstrate How to Use Different Escape Sequences.
- Understand and Use Verbatim/Literal Strings.

CODE ALONG!

END MODULE 9

- Understand How to Format String Values.
- Demonstrate How to Use Different Escape Sequences.
- Understand and Use Verbatim/Literal Strings.

Homework:

1. Quizlet Vocabulary
2. Read Chapter 4

Module 10: Arrays

Arrays – Objectives

- Learn About The Basic Class That Allows the Collection of Other Objects.
- Understand How This Class Helps Apply Activity to Groups of Objects vs. One at a Time.
- The Benefits of This Process.
- Understand the Basic Properties of the Collection Type (Arrays).

Arrays –
Key Terms

- **Collection –** Multiple items grouped together, often of a similar or identical datatype.
- **Array –** A type-safe collection with a fixed-length.
- **Index –** The 0-based position of an item in a collection.
- **Length –** The 1-based total number of items in an array. The Length is always 1 greater than the final index in the array.

Arrays –
Indexes

Dresser		
		String[]
Indexes: 0-based counting	dresser[0]	“tshirts”
	dresser[1]	“pants”
	dresser[2]	“shorts”
	dresser[3]	“socks”

CODE ALONG!

LAB!

Array Lab

LAB!

Input Lab

QUIZ!

Module 8-10 Quiz in Canvas

END MODULE 10

Homework:

1. Quizlet Vocabulary
2. Complete any unfinished labs

- Learn About The Basic Class That Allows the Collection of Other Objects.
- Understand How This Class Helps Apply Activity to Groups of Objects vs. One at a Time.
- The Benefits of This Process.
- Understand the Basic Properties of the Collection Type (Arrays).

Module 11:
Branching with If and Switch

Branching – Objectives

- Discuss the concept of flow control
- Understand when to use branching logic
- Demonstrate how to implement an If Tree
- Demonstrate how to implement a Switch

Branching – Key Terms

- **Branching –** A type of flow control used to make decisions on whether a block of code should run.
- **Ternary Operator –** A quick, single ling if / else statement.
- **Case** – A condition to check used in a switch statement.
- **Break –** Used in a switch statement to tell the compiler to jump out of the switch and continue on with code below it.

Branching –
Intro to Flow Control

- Branching –

  How decisions are made

  - If (Ranges)
  - Switch (exact matching)

- Looping –

  When code needs to be repeated

  - For (count)
  - While (Condition 0-?)
  - Do While (Condition 1-?)
  - Foreach (collections)

Branching –
IF Trees

```
if (condition)
{
    //Code to run
}
else if (condition)
{
    //Code to run
}
else
{
    //Code to run
}
```

- Only one block of code will run
- Executes the first true condition
- “else” will run if nothing else above did
- If trees are good for ranges

CODE ALONG!

Branching – Switch - Case

```
switch (switch-on)
{
    case A:
        //code
        break;
    case B:
        //code
        break;
    default:
        break;
}
```

- Only one block of code will run
- Executes the first matching case.
- “default” will run if nothing else above did
- Switches are good for exact matching

CODE ALONG!

END MODULE 11

- Discuss the concept of flow control
- Understand when to use branching logic
- Demonstrate how to implement an If Tree
- Demonstrate how to implement a Switch

Homework:

1. Quizlet Vocabulary
2. Read Chapter 5

Module 12: Looping

Looping –
Objectives

- Understand the different types of loops in C# and when to use them.
- Demonstrate how to implement each type of loop.
- Discuss how to decide which type of loop to use.

Looping –
3 C's of looping

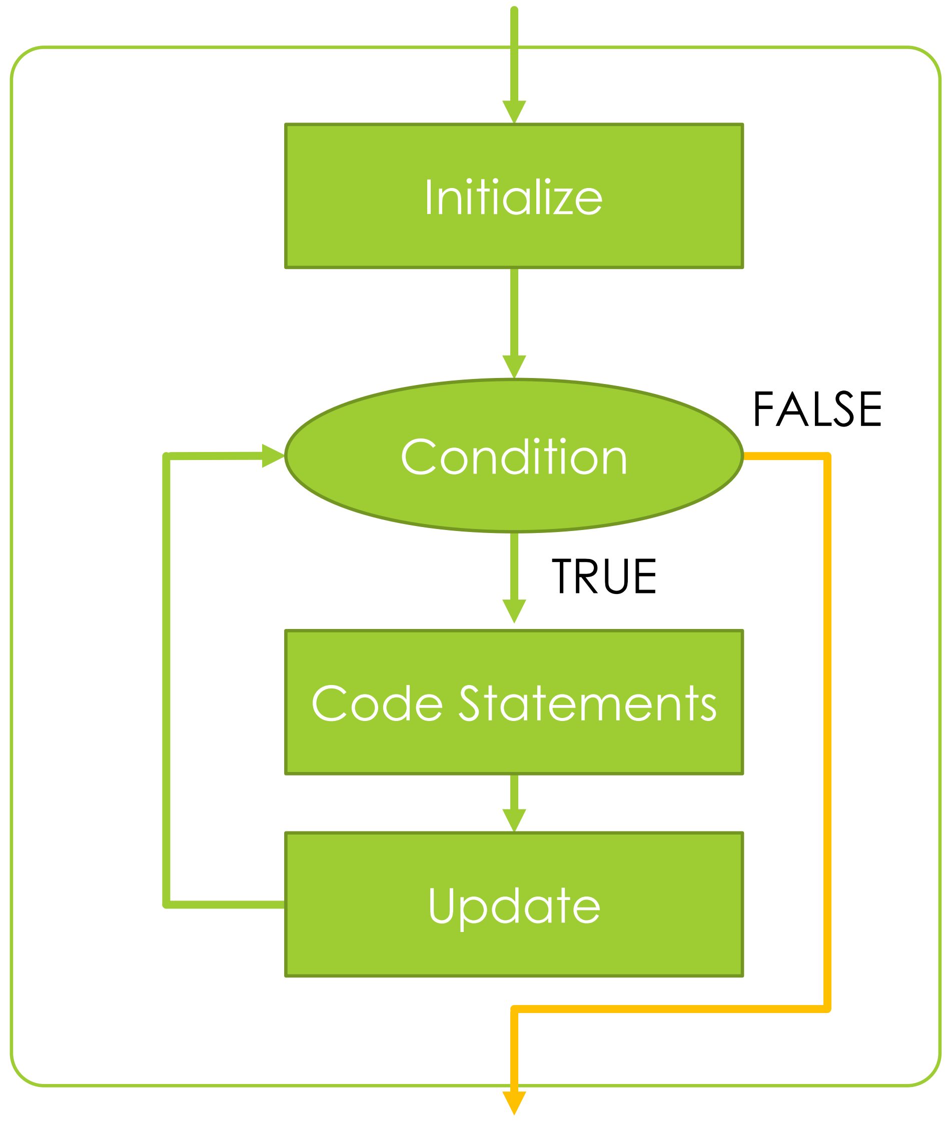
- Count – how many times
- Condition – what triggers the exit
- Collections – the compiler handles it

Looping –
For Loop

```
for (counter; condition; update)
{
	//code to run
}
```

→ (initializer; condition; update) (ICU)

NOTE: for when you know the count or how many times the code needs to repeat



CODE ALONG!

Looping –
While Loop

- Code will run an unknown number of times.

```
counter;
while (condition)
{
    //code to run
    update;
}
```

CODE ALONG!

Looping –
Do While Loop

```
counter;
do
{
    //code to run
    update;
} while (condition)
```

- Code will run at least once and then an unknown number of times.

CODE ALONG!

Looping – Foreach Loop

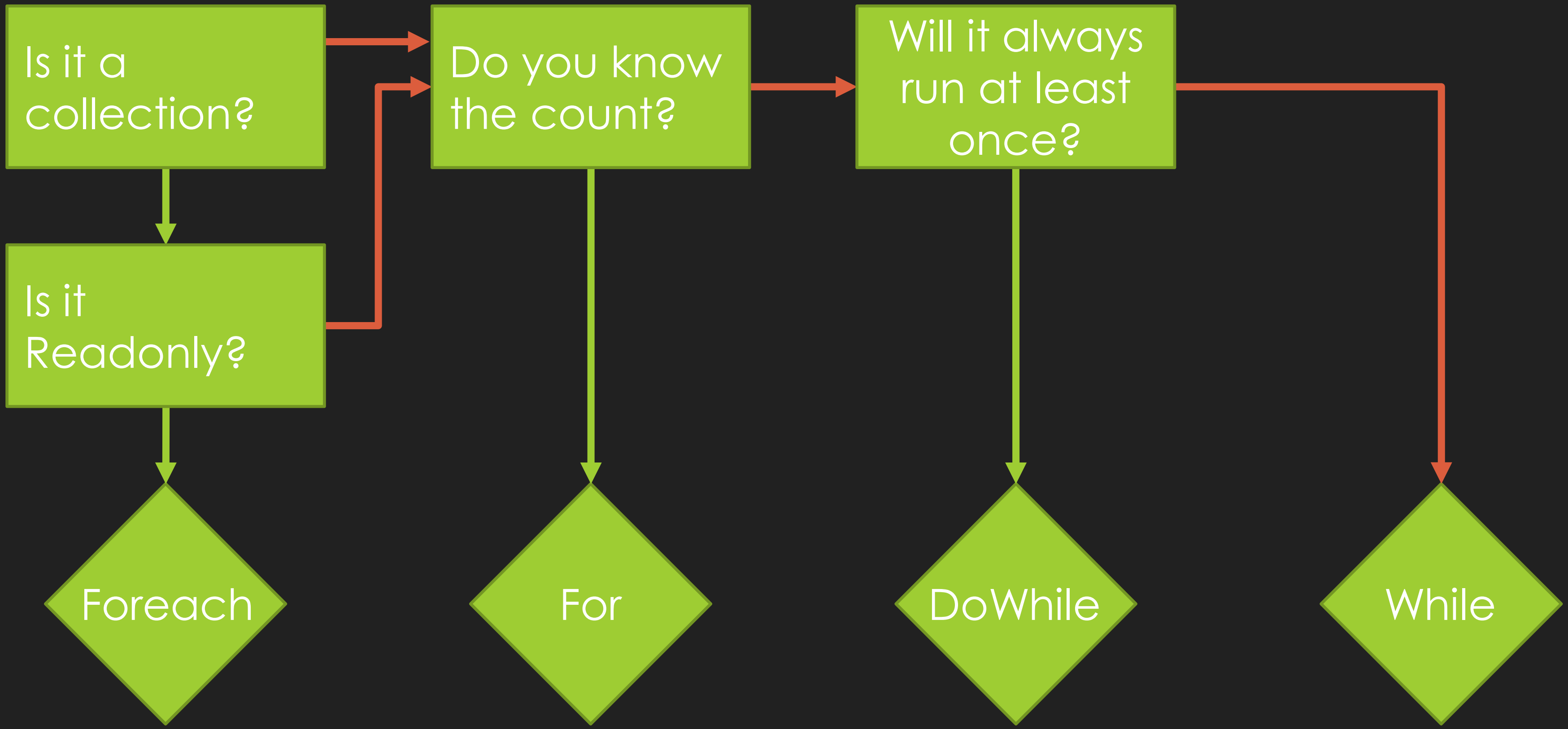
- Only for Collections.
- Read Only Access

```
foreach (var item in collection)
{
    //code to run
}
```

CODE ALONG!

Looping

Which Loop to Use



QUIZ!

Module 11-12 Quiz in Canvas

END MODULE 12

- Understand the different types of loops in C# and when to use them.
- Demonstrate how to implement each type of loop.
- Discuss how to decide which type of loop to use.

Homework:

1. Quizlet Vocabulary
2. C# Fundamentals 1 Homework Packet