**FULL STACK C# .NET LABS**

**UNIT 2: FUNDAMENTAL C#**

[LABS 3](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.8320sz94upfd)

[ROOM CALCULATOR 4](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.h3v3oje82ccv)

[Objectives: Strings, Numbers, Calculations, Input 4](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.cznta9j5gj2c)

[NUMBER ANALYZER - Decision Maker 6](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.gphyi5tnzptt)

[Objectives: User Input, Conditionals 6](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.tlzdsdavsli5)

[EXPONENTS / Powers Table 8](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.afc82ttwxrw0)

[Objectives: User input, calculations 9](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.bpprqqljcrng)

[CAPSTONE: Pig Latin 11](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.whvpqija33)

[Objectives: User Input, String Manipulation 11](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.jk6hw1j2d3i6)

[FACTORIAL CALCULATOR 13](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.hye6qcpm956z)

[Objectives: User Input, Calculations 13](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.w76doimq35ro)

[DICE ROLLER 14](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.s1sqfg3mk1t2)

[Objective: Random Numbers 14](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.z36ja9i7xlrw)

[EXERCISES: Strings, Numbers, Console 17](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.jq87lvgaikwv)

[EXERCISE 1 17](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.942wvsxwsfcs)

[EXERCISE 2 17](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.fj8yy1rhd3md)

[EXERCISE 3 17](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.pni7le9rlbcw)

[EXERCISE 4 18](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.h24g4rd98kzr)

[EXERCISE 5 18](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.5f2ki5h2zy3z)

[EXERCISES: Loops, Expressions, and Control 18](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.m10iqd3iipwg)

[EXERCISE 6 18](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.6kqt5p6ji7c6)

[EXERCISE 7 20](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.nz02agbpz2sb)

[EXERCISE 8 20](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.z2s1cfkpmsp)

[EXERCISE 9 20](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.spxriphrs1ol)

[EXERCISE 10 21](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.cwircccs1k5c)

[EXERCISE 11 21](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.dotci6trp5or)

[EXERCISE 12 22](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.fkrn7nje4rol)

[EXERCISE 13 22](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.la2vuv8eut5u)

[EXERCISE 14 23](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.5sx5jhquxd7p)

[EXERCISE 15 23](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.2dlnzqwu5bti)

[EXERCISE 16 23](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.z4o0vzpf87q6)

[EXERCISE 17 24](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.cpin9gmwc7fa)

[EXERCISE 18 25](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.sa8thbg63kg)

[EXERCISE 19 25](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.rz3rgvd4j0hl)

[EXERCISE 20 26](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.gtjz9jtq9hfq)

[EXERCISE 21 26](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.vwzwac4ufxsi)

[EXERCISE 22 27](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.ckdpu97kzv2q)

[EXERCISES: String Processing 28](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.bvbj4uyxoktl)

[EXERCISE 23 28](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.hdyircxs3p4f)

[EXERCISE 24 29](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.8b6blt58cku4)

[EXERCISE 25 29](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.1qjtt7957z9b)

[EXERCISE 26 30](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.5y9ul2b7qq52)

[EXERCISE 27 30](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.6mwhwwpueh5k)

[EXERCISE 28 30](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.lcjfnmw9kjli)

[EXERCISE 29 31](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.cw1e5amillv9)

[EXERCISE 30 31](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.ruwfpjqnzgyp)

[EXERCISES: Arrays 32](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.9zp2fr3cgw67)

[EXERCISE 31 32](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.o8svcvbt5btc)

[EXERCISE 32 32](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.f6w7ct4by69m)

[EXERCISE 33 33](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.imr1uiigqq5b)

[EXERCISE 34 33](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.nsydr2ptkyqz)

[EXERCISE 35 34](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.ekzul8t62ruz)

[EXERCISE 36 34](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.jhkbw2ptka8f)

[EXERCISE 37 35](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.jl2metugxmcl)

[EXERCISE 38 35](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.dhfsjcs58ueg)

[EXERCISE 39 36](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.2hmsjkk1bdrk)

[EXERCISE 40 37](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.qyvbfe4ph02s)

[EXERCISE 41 37](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.7in0gj5foxtx)

[BONUS LABS 39](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.mwdobjpilgs2)

[BONUS-1: Number Grades to Letter Grades 40](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.u2cvdgcirr3d)

[BONUS-2: Age Calculator 41](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.38flrnyeuoyk)

[BONUS-3: Random Number Guesser 42](https://docs.google.com/document/d/16ZeIEf9UaWFY32P8jz_PFbBm1JSKzUoP1lte9PfWrS4/preview#heading=h.b4qk3kfx0f11)

**LABS**

**ROOM CALCULATOR**

**Objectives: Strings, Numbers, Calculations, Input**

**NOTE: Points will still be awarded for items that are written correctly but don’t actually work due to other things being broken. There are a total of 10 points available for this lab.**

**Task**: Calculate the perimeter and area of various rooms.

**What will the application do?**

* **1 Point** The application prompts the user to enter values of length and width of the classroom.
* **2 Points:** The application displays the area and perimeter of that classroom.

**Build Specifications:**

1. **1 Point:** Assume that the rooms are perfect rectangles.
2. **1 Point:** Assume that the user will enter valid numeric data for length and width.
3. **1 Point:** The application should accept decimal entries.

**Additional Requirements:**

* **1 Point:** For answering the Lab Summary when submitting to the LMS
* **-2 Points:** if there are any syntax errors or if the program does not run (for example, in a Main method).

**Hints:**

* Don’t mess up the formulas for area or perimeter.
* The Snug is 24’ 6” x 20’ 0”. The Penthouse is 42’ 6” x 16’ 6”.

**Extra Challenges:**

* **2 Points:** The application classifies the room as small (up to 250 square feet), medium (more than 250 but less than 650 square feet, or large (650 square feet or more).
* **1 Point:** Calculate the volume of the rooms.
* **1 Point:** Calculate the surface area of the rooms.

**Console Preview:**

|  |
| --- |
| Welcome to Grand Circus’ Room Detail Generator!  Enter Length: {user input here, for example: 24.5}  Enter Width: {user input here, for example: 20}  Area: {calculated: 490}  Perimeter: {calculated: 89 }  {calculated: This is a medium-sized room.}  Thank you for using the Room Detail Generator! |

**NUMBER ANALYZER - Decision Maker**

**Objectives: User Input, Conditionals**

**NOTE: Points will still be awarded for items that are written correctly but don’t actually work due to other things being broken. There are a total of 10 points available for this lab.**

**Task**: Useconditional statements to automate the decision-making process.

**What will the application do?**

* **1 Point:** The application prompts the user to enter an integer between 1 and 100.
* Display the associated result based on the integer range entered.

**Build Specifications:**

1. **1 Point:** Use if/else statements to take different actions depending on user input.
2. Given an integer entered by a user, perform the following conditional actions:
   1. **1 Point:** If the integer entered is odd and less than 60, print the number entered and “Odd and less than 60.”
   2. **1 Point:** If the integer entered is even and in the inclusive range of 2 to 24, print “Even and less than 25.”
   3. **1 Point:** If the integer entered is even and in the inclusive range of 26 to 60, print “Even and between 26 and 60 inclusive.”
   4. **1 Point:** If the integer entered is even and greater than 60, print the number entered and “Even and greater than 60.”
   5. **1 Point:** If the integer entered is odd and greater than 60, print the number entered and “Odd and greater than 60.”

**Additional Requirements:**

* **1 Point:** For answering the Lab Summary when submitting to the LMS
* **-2 Points:** if there are any syntax errors or if the program does not run (for example, in a Main method).

**Hints:**

* Remember what “inclusive” and “exclusive” mean when referring to ranges of numbers. The range of numbe0
* rs from 1 inclusive to 10 exclusive means the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9 (but not 10; it’s excluded).

**Extra Challenges (2 Points Maximum):**

* **1 Point:** Set up the program to continue running with a prompt at the end to see if they want to stop. (hint: Loops)
* **1 Point:** Ask for user information (ex. name) at the beginning of the application, and use it to refer to the user throughout the application.
* **1 Point:** Add validation to guarantee that a user enters a positive integer between 1 and 100.

**Console Preview:**

|  |
| --- |
| **Enter a number between 1 and 100:** {user input here, for example: 3}  **Output:** {output here, 3 and Odd }  Bye! |

**EXPONENTS / Powers Table**

**Objectives: User input, calculations**

**NOTE: Points will still be awarded for items that are written correctly but don’t actually work due to other things being broken. There are a total of 10 points available for this lab.**

**Task**: Display a table of powers. Ask the user if they would like to go again, and if so loop back to the beginning.

**What will the application do?**

* **1 Point:** The application prompts the user to enter an integer.
* **3 Point:** The application displays a table of squares and cubes from 1 to the value entered.
* **1 Point:** The application prompts the user to continue.

**Build Specifications:**

* **1 Point:** Assume that the user will enter valid data.
* **1 Point:** The application should continue only if the user agrees to.

**Additional Requirements:**

* **1 Point:** For answering the Lab Summary when submitting to the LMS
* **-2 Points:** if there are any syntax errors or if the program does not run (for example, in a Main method).

**Hints:**

* Don’t mess up the difference between squares and cubes!
* Use \t to tab to line up columns properly
* Your instructor will provide you with sample loops during class regarding how to ask the user if they would like to go again.

**Extra Challenges:**

* **1 Point:** Provide validation by rejecting 0 or negative numbers as user input; keep prompting the user until they enter a valid number.
* **1 Point:** Research formatted strings and right-align the numbers in columns instead of left-aligning them
* **1 Point:** Find out the maximum number whose cube will fit in an int, and limit the user input to that number or less

**Console Preview:**

|  |
| --- |
| Learn your squares and cubes!  Enter an integer: {user input here, for example: 5}  Number      Squared        Cubed  =======   ======= ======  1 1 1  2 4 8  3          9             27  4       16           64  5              25           125  Continue? (y/n): {user input here, for example: Y}  Enter an integer: … |

**CAPSTONE: Pig Latin**

**Objectives: User Input, String Manipulation**

NOTE: Points will be awarded for items that are written correctly in themselves but don’t actually work because other things are broken. There is a total of 10 points available for this lab.

**Intro:** Pig Latin is a children’s word game in English where starting consonants are flipped to the ends of words and -ay is added to each word.  Hello World would be elloyhay orldway in Pig Latin, for instance.  Many languages have games similar to this--read more at <http://mentalfloss.com/article/50242/pig-latins-11-other-languages>

**Task:** Translate from English to Pig Latin

**What will the application do?**

* **1 Point:** The application prompts the user for a word.
* **1 Point:** The application translates the text to Pig Latin and displays it on the console.
* **1 Point:** The application asks the user if he or she wants to translate another word.

**Build Specifications:**

* **1 Point:** Convert each word to a lowercase before translating.
* **1 Point:** If a word starts with a vowel, just add “way” onto the ending.
* **2 Point**: if a word starts with a consonant, move all of the consonants that appear before the first vowel to the end of the word, then add “ay” to the end of the word.

**Additional Requirements:**

* **1 Point:** For answering Lab Summary when submitting to the LMS
* **-2 Points:** if there are any syntax errors or if the program does not run (for example, in a Main method).

**Extended Exercises (2 points maximum):**

* **1 Point:** Keep the case of the word, whether its uppercase (WORD), title case (Word), or lowercase (word).
* **1 Point:** Allow punctuation in the input string.
* **1 Point:** Translate words with contractions. Ex: can’t become an’tcay
* **1 Point:** Don’t translate words that have numbers or symbols. Ex: 189 should be left as 189 and hello@grandcircus.co should be left as [hello@grandcircus.co](mailto:hello@grandcircus.co).
* **1 Point:** Check that the user has actually entered text before translating.

**1 Point:** Make the application take a line instead of a single word, and then find the Pig Latin for each word in the line.

**Hints:**

* Treat “y” as a consonant.

**Console Preview**

|  |
| --- |
| Welcome to the Pig Latin Translator!  Enter a line to be translated: {this sentence exists here}  Isthay entencesay existsway erehay  Translate another line? (y/n): N |

**FACTORIAL CALCULATOR**

**Objectives: User Input, Calculations**

**Task**: Calculate the factorial of a number.

**What will the application do?**

* The application prompts the user to enter an integer from 1 to 10.
* The application displays the factorial of the number entered by the user.
* The application prompts the user to continue.

**Build Specifications:**

1. Use a for loop to calculate the factorial.
2. Assume that the user will enter valid data.
3. Use the long type to store the factorial.
4. The application should continue only if the user agrees to.

**Hints:**

* A factorial is a number multiplied by each of the numbers before it. Factorials are denoted by the exclamation point (n!). Ex:
  + 1! = 1 which equals 1
  + 2! = 1 x 2 which equals 2
  + 3! = 1 x 2 x 3 which equals 6
  + 4! = 1 x 2 x 3 x 4 which equals 24

**Extra Challenges:**

* Test the application and find the integer for the highest factorial that can be accurately calculated by this application, then modify the prompt so that it prompts the user for a number “from 1 to {the highest integer that returns accurate factorial calculation}”.
* Use Recursion to implement the factorial.

**Tests:**

* Test 1

**Console Preview:**

|  |
| --- |
| Welcome to the Factorial Calculator!  Enter an integer that’s greater than 0 but less than 10: {user input here, for example: 3}  The factorial of 3 is 6.  Continue? (y/n): {user input here, for example: Y}  Enter an integer that’s greater than 0 but less than 10: {user input here, for example: 9}  The factorial of 9 is 362880.  ... |

**DICE ROLLER**

**Objective: Random Numbers**

**Task**: Create an application that simulates dice rolling.

**What will the application do?**

* The application asks the user to enter the number of sides for a pair of dice.
  + If you have learned about exception handling, make sure the user can only enter numbers
* The application prompts the user to roll the dice.
* The application “rolls” two n-sided dice, displaying the results of each along with a total
* For 6-sided dice, the application recognizes the following dice combinations and displays a message for each. It should not output this for any other size of dice.
  + Snake Eyes: Two 1s
  + Ace Deuce: A 1 and 2
  + Box Cars: Two 6s
  + Win: A total of 7 or 11
  + Craps: A total of 2, 3, or 12 (will also generate another message!)
* The application asks the user if he/she wants to roll the dice again.

**Build Specifications:**

1. Create a static method to generate the random numbers.
   1. Proper method header: 2 points
   2. Program generates random numbers validly within the user-specified range: 1 point
   3. Method returns meaningful value of proper type: 1 point
2. Create a static method for six-sided dice that takes two dice values as parameters, and returns a string for one of the valid combinations (e.g. Snake Eyes, etc.) or an empty string if the dice don’t match one of the combinations.
3. Create a static method to implement output for different dice combinations
   1. Proper method header: 2 points
   2. Method recognizes all specified cases correctly: 1 point
   3. Method displays properly to Console: 1 point
4. Application takes all user input correctly: 1 point
5. Application loops properly: 1 point

**Hints:**

* Use the Random class to generate a random number.

**Extra Challenges:**

* Come up with a set of winning combinations for other dice sizes besides 6.

**Console Preview:**

|  |
| --- |
| Welcome to the Grand Circus Casino!  How many sides should each die have?  {6}  Roll 1:  You rolled a 2 and a 5 (7 total)  Win!  Roll again? (y/n): {y}  Roll 2:  You rolled a 6 and a 6 (12 total)  Boxcars  Craps!  Roll again? (y/n): {y}  Roll 2:  You rolled a 3 and a 5 (8 total)  Roll again? (y/n): {n}  Thanks for playing!! |

**EXERCISES: Strings, Numbers, Console**

**EXERCISE 1**

**Description**  
Prompt the user to enter a string. After the user enters a string, output the same string back to the console.  
  
**Example**  
>>Enter some text: <<Hello, World! ECHOOOOOO!  
>>Hello, World! ECHOOOOOO!

**EXERCISE 2**

**Description**

Prompt the user to enter a number. After the user enters a number, add 1 to the number and output it back to the console.

**Example**

>>Enter a number: <<52

>>53

**EXERCISE 3**

**Description**

Prompt the user to enter a number. After the user enters a number, add .5 to the number and output it back to the console.

**Example**

>>Enter a number: <<17.3

>>17.8

**EXERCISE 4**

**Description**

Prompt the user to enter two numbers. After the user enters the numbers, add them together and output the sum back to the console.

**Example**

>>Enter a number: <<12.2

>>Enter another number: <<17.3

>>The sum is 29.5

**EXERCISE 5**

**Description**

Prompt the user to enter two numbers. After the user enters the numbers, multiply them and output the product back to the console.

**Example**

>>Enter a number: <<10.2

>>Enter another number: <<13.4

>>The product is 136.68

**EXERCISES: Loops, Expressions, and Control**

**EXERCISE 6**

**Description**

Use a do-while loop to output "Hello, World!" in a loop. Each time you output "Hello, World!" ask the user whether they would like to continue.

**Example**

>>Hello, World!

Would you like to continue (y/n)? <<y

>>Hello, World!

Would you like to continue (y/n)? <<y

>>Hello, World!

Would you like to continue (y/n)? <<y

>>Hello, World!

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 7**

**Description**

Use a do-while loop to run exercise 1 in a loop. Each time you run the code ask the user whether they would like to continue.

**Example**

>>Enter some text: <<Hello, World! ECHOOOOOO!

>>Hello, World! ECHOOOOOO!

Would you like to continue (y/n)? <<y

>>Enter some text: <<Hello, World! ECHOOOOOO again!

>>Hello, World! ECHOOOOOO again!

Would you like to continue (y/n)? <<y

>>Enter some text: <<Hello, World! ECHOOOOOO againnnn!

>>Hello, World! ECHOOOOOO againnnn!

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 8**

Make exercises 2-5 run in a loop. Use a do-while loop to run the code in a loop. Each time you run the code ask the user whether they would like to continue.

**EXERCISE 9**

**Description**

Prompt the user to enter a language.  Based on the language the user input, display "Hello, World!" in that language.  Use a switch statement to choose what to display.

**Example**

>>Enter a language: <<English

>>Hello, World!

Would you like to continue (y/n)? <<y

>>Enter a language: <<Spanish

>>Hola Mundo!

Would you like to continue (y/n)? <<y

>>Enter a language: <<Dutch

>>Hallo wereld!

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 10**

**Description**

Determine whether the user is tall enough to ride a roller coaster. Prompt the user to enter her height in inches.  If she is less than 54 inches tall, notify her that she cannot ride the Raptor. If she is at least 54 inches tall, notify her that she can ride the Raptor.

**Example**

>>Enter your height in inches: 52.3

>>Sorry, you cannot ride the Raptor. You need 1.7 more inches.

Would you like to continue (y/n)? <<y

>>Enter your height in inches: 55.9

>>Great, you can ride the Raptor!

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 11**

**Description**

Use a for loop to output all the numbers from 0 to 9.

**Example**

>>0 1 2 3 4 5 6 7 8 9

Would you like to continue (y/n)? <<y

>>0 1 2 3 4 5 6 7 8 9

Would you like to continue (y/n)? <<y

>>0 1 2 3 4 5 6 7 8 9

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 12**

**Description**

Output all the numbers from 9 to 0. Use a for loop to output all the numbers from 9 to 0.

**Example**

>>9 8 7 6 5 4 3 2 1 0

Would you like to continue (y/n)? <<y

>>9 8 7 6 5 4 3 2 1 0

Would you like to continue (y/n)? <<y

>>9 8 7 6 5 4 3 2 1 0

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 13**

**Description**

Prompt the user for a number. Use a for loop to output all the numbers from that number to 0.

**Example**

>>Enter a number: <<5

>>5 4 3 2 1 0

Would you like to continue (y/n)? <<y

>>Enter a number: <<12

>>12 11 10 9 8 7 6 5 4 3 2 1 0

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 14**

**Description**

Prompt the user for a number. Use a for loop to output the squares of all the numbers from 1 to that number.

**Example**

>>Enter a number: <<2

>>1 4

Would you like to continue (y/n)? <<y

>>Enter a number: <<7

>>1 4 9 16 25 36 49

Would you like to continue (y/n)? <<n

**EXERCISE 15**

**Description**

Prompt the user for a number. Use a for loop to output the cubes of all the numbers from 1 to that number.

**Example**

>>Enter a number: <<2

>>1 8

Would you like to continue (y/n)? <<y

>>Enter a number: <<7

>>1 8 27 64 125 216 343

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 16**

**Description**

Use a for loop to output a triangle of asterisks with a height of ten.

**Example**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

**EXERCISE 17**

**Description**

Use a for loop to output a triangle of asterisks with a height of ten.

**Example**

1. \*
2. \* \*
3. \* \* \*
4. \* \* \* \*
5. \* \* \* \* \*
6. \* \* \* \* \* \*
7. \* \* \* \* \* \* \*
8. \* \* \* \* \* \* \* \*
9. \* \* \* \* \* \* \* \* \*
10. \* \* \* \* \* \* \* \* \* \*

**EXERCISE 18**

**Description**

Prompt the user to enter a number.  Use a for-loop to calculate the sum of all the numbers from 1 to the number entered.

**Example**

>>Enter a number: <<100

>>5050

Would you like to continue (y/n)? <<y

>>Enter a number: <<20

>>210

Would you like to continue (y/n)? <<n

<<Goodbye!

**EXERCISE 19**

**Description**

Prompt the user to enter two numbers.  Use a for-loop to calculate the sum of all the numbers from the first number entered to the second.

**Example**

>>Enter a number: <<12

>>Enter another number: <<21

<<The sum of all the numbers from 12 to 21 is 165.

Would you like to continue (y/n)? <<y

>>Enter a number: <<3

>>Enter another number: <<5

<<The sum of all the numbers from 3 to 5 is 12.

Would you like to continue (y/n)? <<n

<<Goodbye!

**EXERCISE 20**

**Description**

Prompt the user to enter a number.  Use a for-loop to calculate the product of the number and the two preceding numbers.

**Example**

>>Enter a number: <<6

>>The product of 6, 5, and 4 is 120.

Would you like to continue (y/n)? <<y

>>Enter a number: <<8

>>The product of 8, 7, and 6 is 120.

Would you like to continue (y/n)? <<n

<<Goodbye!

**EXERCISE 21**

**Description**

Prompt the user to enter a series of words.  Once the user is done entering the words, output a sentence containing all the words.

**Example**

>>Enter a word: <<The

>>Would you like to enter another word (y/n)? <<y

>>Enter a word: <<cow

>>Would you like to enter another word (y/n)? <<y

>>Enter a word: <<jumped

>>Would you like to enter another word (y/n)? <<y

>>Enter a word: <<over

>>Would you like to enter another word (y/n)? <<y

>>Enter a word: <<the

>>Would you like to enter another word (y/n)? <<y

>>Enter a word: <<moon.

>>Would you like to enter another word (y/n)? <<n

>>The cow jumped over the moon.

Would you like to continue (y/n)? <<y

>>Enter a word: <<Hello,

>>Would you like to enter another word (y/n)? <<y

>>Enter a word: <<World!

>>Would you like to enter another word (y/n)? <<

>>Hello, World!

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 22**

**Description**

Prompt the user to enter two numbers that will determine a range.  Then prompt the user to enter another number and check if it is in the range.

**Example**

>>Enter a number: <<7

>>Enter another number: <<25

>>Your range is 7-25.

Enter a number to verify it is in the range: <<20

>>20 is in the range.

Would you like to continue (y/n)? <<y

>>Enter a number to verify it is in the range: <<32

>>32 is outside the range.

Would you like to continue (y/n)? <<y

>>Enter a number to verify it is in the range: <<7

>>7 is in the range.

Would you like to continue (y/n)? <<n

<<Goodbye!

**EXERCISES: String Processing**

**EXERCISE 23**

**Description**

Prompt the user to enter a string.  Extract and output the first ten characters of the string.

**Example**

>>Enter some text: <<abcdefghijklmnop

<<The first ten characters were: abcdefghij

Would you like to continue (y/n)? <<y

>>Enter some text: <<Hello, World!

<<The first ten characters were: Hello, Wor

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 24**

**Description**

Prompt the user to enter a string.  Extract and output the last ten characters of the string.

**Example**

>>Enter some text: <<abcdefghijklmnop

<<The last ten characters were: ghijklmnop

Would you like to continue (y/n)? <<y

>>Enter some text: <<Hello, World!

<<The last ten characters were: lo, World!

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 25**

**Description**

Prompt the user to enter a sentence.  Split the sentence into individual words and display each word on its own line.

**Example**

>>Enter a sentence: <<The cow jumped over the moon.

>>The

cow

jumped

over

the

moon.

Would you like to continue (y/n)? <<y

>>Enter a sentence: <<Hello, World!

>>Hello,

World!

Would you like to continue (y/n)? <n

<<Goodbye!

**EXERCISE 26**

**Description**

Prompt the user to enter text.  Count and output how many vowels are in the string.

**Example**

>>Enter some text: <<abcdefghijklmnopqrstuvwxyz

>>There were 5 vowels.

Would you like to continue (y/n)? <<y

>>Enter some text: <<Hello, World!

>>There were 3 vowels.

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 27**

**Description**

Prompt the user to enter text.  Count and output how many consonants are in the string.

**Example**

>>Enter some text: <<abcdefghijklmnopqrstuvwxyz

>>There were 19 consonants.

Would you like to continue (y/n)? <<y

>>Enter some text: <<Hello, World!

>>There were 7 consonants.

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 28**

**Description**

Prompt the user to enter text.  Remove all the vowels and output the text.

**Example**

>>Enter some text: <<abcdefghijklmnopqrstuvwxyz

>>bcdfghjklmnpqrstvwxyz

Would you like to continue (y/n)? <<y

>>Enter some text: <<Hello, World!

>>Hll, Wrld!

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 29**

**Description**

Prompt the user to enter text.  Remove all the vowels in the middle of the word, but leave any that start or end the word.

**Example**

>>Enter some text: <<Elephants are wonderful!

>>Elphnts are wndrfl!

Would you like to continue (y/n)? <<y

>>Enter some text: <<Is every flake edible?

>>Is evry flke edble?

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 30**

**Description**

Prompt the user to enter text.  Reverse the text.

**Example**

>>Enter some text: <<abcdefghijklmnopqrstuvwxyz

>>zyxwvutsrqponmlkjihgfedcba

Would you like to continue (y/n)? <<y

>>Enter some text: <<Hello, World!

>>!dlroW ,olleH

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISES: Arrays**

**EXERCISE 31**

**Description**

Create an array of size 5 and fill it with the following numbers: 2, 8, 0, 24, 51. Prompt the user to enter an index. Display the element in the array at that index.

**Example**

>>Enter an index of the array: <<2

>>The value at index 2 is 0.

Would you like to continue (y/n)? <<y

>>Enter an index of the array: <<7

<<That is not a valid index.

Would you like to continue (y/n)? <<y

>>Enter an index of the array: <<0

>>The value at index 0 is 2.

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 32**

**Description**

Create an array of size 5 and fill it with the following numbers: 2, 8, 0, 24, 51. Prompt the user to enter a number. If the number is in the array, display the index at which it is located.

**Example**

>>Enter a number: <<12

>>That value cannot be found in the array.

Would you like to continue (y/n)? <<y

>>Enter an index of the array: <<24

<<The value 24 can be found at index 3.

Would you like to continue (y/n)? <<y

>>Enter an index of the array: <<abc

>>That value cannot be found in the array.

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 33**

**Description**

Create an array of size 5 and fill it with the following numbers: 2, 8, 0, 24, 51. Let the user change the array by specifying an index and a replacement number.

**Example**

>>Enter an index of the array: <<2

>>The value at index 2 is 0.  Would you like to change it (y/n)? <<y

>>Enter the new value at index 2: <<17

>>The value at index 2 is 17.  Would you like to continue (y/n)? <<y

>>Enter an index of the array: <<2

>>The value at index 2 is 17.  Would you like to change it (y/n)? <<n

>>Would you like to continue (y/n)? <<y

>>Enter an index of the array: <<7

<<That is not a valid index. Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 34**

**Description**

Create an array of size 5 and fill it with the following numbers: 16, 32, 64, 128, 256. Prompt the user to enter a command, 'half' or 'double'.  If the user enters 'half', half all the elements in the array.  If the user enters 'double', double all the elements in the array.

**Example**

>>Enter a command (half/double): half

>>The array now contains: 8, 16, 32, 64, 128.

Would you like to continue (y/n)? <<y

>>Enter a command (half/double): half

>>The array now contains: 4, 8, 16, 32, 64.

Would you like to continue (y/n)? <<y

>>Enter a command (half/double): double

>>The array now contains: 8, 16, 32, 64, 128.

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 35**

**Description**

Create an array of size 5 and fill it with the following strings: "cow", "elephant", "jaguar", "horse", "crow". Prompt the user to enter two indices separated by a space.  The first index will specify the word, and the second will specify a letter in that word.  Display the corresponding word and letter.

**Example**

>>Enter two indices: <<2 0

>>The value at index 2 is jaguar.  The letter at index 0 is j.

Would you like to continue (y/n)? <<y

>>Enter two indices: <<4 2

>>The value at index 4 is crow.  The letter at index 2 is o.

Would you like to continue (y/n)? <<y

>>Enter two indices: <<0 1

>>The value at index 0 is cow.  The letter at index 1 is o.

Would you like to continue (y/n)? <<y

>>Enter two indices: <<7 0

>>Those are not valid indices.

Would you like to continue (y/n)? <<y

>>Enter two indices: <<2 17

>>Those are not valid indices.

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 36**

**Description**

Create two arrays, each of size 5. Fill the first array with the numbers: 12, 11, 10, 9, 8. Fill the second array with the strings: "Drummers Drumming", "Pipers Piping", "Lords a-Leaping", "Ladies Dancing", "Maids a-Milking". Combine both arrays to display a piece of the holiday song.

**Example**

>>Enter a command (sing/quit): sing

>>12 Drummers Drumming

11 Pipers Piping

10 Lords a-Leaping

9 Ladies Dancing

8 Maids a-Milking

Enter a command (sing/quit): quit

>>Goodbye!

**EXERCISE 37**

**Description**

Prompt the user to enter five numbers.  Store these numbers in an array and output their sum.

**Example**

>>Enter a number: <<7

>>Enter a number: <<12

>>Enter a number: <<45

>>Enter a number: <<29

>>Enter a number: <<12

>>7 + 12 + 45 + 29 + 12 = 105

<<Would you like to continue (y/n)? <<y

>>Enter a number: <<11

>>Enter a number: <<78

>>Enter a number: <<5

>>Enter a number: <<0

>>Enter a number: <<30

>>11 + 78 + 5 + 0 + 30 = 124

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 38**

**Description**

Prompt the user to enter five numbers.  Store these numbers in an array and output their average.

**Example**

>>Enter a number: <<7

>>Enter a number: <<12

>>Enter a number: <<45

>>Enter a number: <<29

>>Enter a number: <<12

>>(7 + 12 + 45 + 29 + 12)/5 = 21

<<Would you like to continue (y/n)? <<y

>>Enter a number: <<11

>>Enter a number: <<78

>>Enter a number: <<5

>>Enter a number: <<0

>>Enter a number: <<30

>>(11 + 78 + 5 + 0 + 30)/5 = 24.8

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 39**

**Description**

Prompt the user to enter five numbers.  Store these numbers in an array and output them in ascending order.

**Example**

>>Enter a number: <<7

>>Enter a number: <<12

>>Enter a number: <<45

>>Enter a number: <<29

>>Enter a number: <<12

>>7 12 12 29 45

<<Would you like to continue (y/n)? <<y

>>Enter a number: <<11

>>Enter a number: <<78

>>Enter a number: <<5

>>Enter a number: <<0

>>Enter a number: <<30

>>0 5 11 30 78

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 40**

**Description**

Prompt the user to enter five numbers.  Store these numbers in an array and output the median.

**Example**

>>Enter a number: <<7

>>Enter a number: <<12

>>Enter a number: <<45

>>Enter a number: <<29

>>Enter a number: <<12

>>The median of (7, 12, 12, 29, 45) is 12.

<<Would you like to continue (y/n)? <<y

>>Enter a number: <<11

>>Enter a number: <<78

>>Enter a number: <<5

>>Enter a number: <<0

>>Enter a number: <<30

>>The median of (0, 5, 11, 30, 78) is 11.

Would you like to continue (y/n)? <<n

>>Goodbye!

**EXERCISE 41**

**Description**

Prompt the user to enter two numbers.  Divide the two numbers and show only two decimal places.

**Example**

>>Enter a number: <<1

>>Enter another number: <<8

>>1/8 is approximately .13.

Would you like to continue (y/n)? <<y

>>Enter a number: <<7

>>Enter another number: <<3

>>7/3 is approximately 2.33

Would you like to continue (y/n)? <<y

>>Enter a number: <<5

>>Enter another number: <<0

>>You cannot divide by 0.

Would you like to continue (y/n)? <<n

>>Goodbye!

**BONUS LABS**

**BONUS-1: Number Grades to Letter Grades**

**Task:** Convert given number grades into letter grades.

**What will the application do?**

* The user will enter a numerical grade from 0 to 100.
* The application will display the corresponding letter grade.
* The application will prompt the user to continue.

**Build Specifications**

1. Assume that the user will enter valid integers for the grades.
2. The application should only continue if the user enters “y” or “Y”.

**Hints:**

* Grade Ranges:
  + A : 100 - 88
  + B : 87 - 80
  + C :  79 - 67
  + D : 66 - 60
  + F : 60 - 0

**Extended Challenge:**

Edit your grade ranges to include pluses and minuses (ex: 99-100 = A+).

**Console Preview:**

|  |
| --- |
| Welcome to the Letter Grade Converter!  Enter a numerical grade: 90  Letter Grade: A  Continue? (y/n): Y  Enter a numerical grade: 82  Letter Grade: B  Continue? (y/n): y  Enter a numerical grade: 51  Letter Grade: F  …. |

**BONUS-2: Age Calculator**

**Task:** Calculate the age of a person!

**What will the application do?**

* Ask the user to enter their birthday.  You can ask the user to enter the year, month, then day, then store it in a DateTime variable.
* Calculate the Age of the user (Show the output in years and days).

**Extra Challenges:**

* If the current day is the user’s birthday, print a nice message!

**BONUS-3: Random Number Guesser**

**Task:** Guess a number from 1 to 100. You should complete this by pair programming.

Yes, *really.* Buddy up, and close one of your computers.

**What will the application do?**

* The application prompts the user for an *int value* from 1 to 100 until the user guesses the random number that the application has generated.
* The application displays messages that indicate whether the user’s guess is too high or too low.
* When the user finally guesses the number, the application displays the number of guesses it took the user along with a rating.
* The application prompts the user to play again.
* When the user chooses to end the game, the application displays a goodbye message.

**Build Specifications**

1. If the user’s guess is more than 10 higher or more than 10 lower than the random number, the user should be notified as such. (ex: “Way too high!”)
2. If the user’s guess is less than 10 higher or less than 10 lower than the random number, then the user should be notified as such. (ex: “Too low!”)
3. The message that’s displayed when the user successfully gets the number should change based on the number of guesses it took him or her.
4. The application should only accept numbers from 1 to 100.
5. When the user is prompted to Play Again, the application should only accept a value of “y” or “n”.
6. If the user enters invalid data, the application should display an error message and prompt the user for data again.
7. The code used to validate the data should be stored in separate methods.

**Hints:**

* Use the Random class to generate a random number.
* Hint for number 7:
  + public static double GetDoubleWithinRange(string Prompt, double Min, double Max)
* public static int GetIntWithinRange(string Prompt, int Min, int Max)

BONUS-3

**Console Preview:**

|  |
| --- |
| Welcome to the Guess a Number Game!  +++++++++++++++++++++++++++++++++  I’m thinking of a number between 1 and 100.  Try to guess it, n00b.  Enter your lousy guess: 8  You got it in 1 tries.  You must be seriously amazing slash telepathic!  Try again? (y/n): y  I’m thinking of a number between 1 and 100.  Try to guess it, n00b.  Enter your lousy guess: 8  You’re crazy low, bro. Try again.  Enter a number: 49  Too low, Joe. Try again.  Enter a number: 56  Too high! Try again.  Enter a number: 51  You got it in 4 tries.  Pretty good, I guess. I bet you can do better, though.  Try again? (y/n): …. |

**BASIC LOOPS LAB**

**Description**

Use a do-while loop to output "Hello, World!" in a loop. Each time you output "Hello, World!" ask the user whether they would like to continue.

**Example**

>>Hello, World!

Would you like to continue (y/n)? <<y

>>Hello, World!

Would you like to continue (y/n)? <<y

>>Hello, World!

Would you like to continue (y/n)? <<y

>>Hello, World!

Would you like to continue (y/n)? <<n

>>Goodbye!

**Description**

Prompt the user for a number. Use a for loop to output all the numbers from that number to 0. After that loop finishes, use another loop to output all the numbers from 0 to that number.

**Example**

>>Enter a number: <<5

>>5 4 3 2 1 0

>>0 1 2 3 4 5

Would you like to continue (y/n)? <<y

>>Enter a number: <<12

>>12 11 10 9 8 7 6 5 4 3 2 1 0

>>0 1 2 3 4 5 6 7 8 9 10 11 12

Would you like to continue (y/n)? <<n

>>Goodbye!

**Description**

A door has a keypad entry. The combination to get in is 13579. Write a while loop (not a do while loop) that asks the user to enter a key code. The loop will repeat as long as the user enters the wrong code. After the user enters the correct code, the program will print out a welcome message.

Hint: There are many ways to accomplish this, but one way is to create a boolean variable that represents if the door is locked or unlocked. Then think about real life, when you approach a door with a keypad, what state is it initially in before you type anything into the keypad?

**Description**

Continue the previous exercise, but now add a limited number of tries, say 5. After 5 unsuccessful attempts, the loop ends, but instead of printing a welcome message, it prints a message warning that there were too many incorrect attempts. (But if the user entered the correct number, it will still print the welcome message as before.)

**Description**

Start a new console project, and repeat the same exercise as the previous, except this time implement it with a do while loop.

**Optional Stretch**

Move the last while loop or the do-while loop into its own function. The function should return a true if access is granted, or a false if the user didn’t enter the correct code within 5 tries.

bool playAgain;

do

{

    Random random = new Random();

    int returnValue = random.Next(1, 101);

    int userGuess = 0;

    Console.WriteLine("I am thinking of a number between 1-100.  Can you guess what it is?");

    while (userGuess != returnValue)

    {

        userGuess = Convert.ToInt32(Console.ReadLine());

        if (userGuess < returnValue)

        {

            Console.WriteLine("No, the number I am thinking of is higher than " + userGuess +

                              ". Can you guess what it is?");

        }

        else if (userGuess > returnValue)

        {

            Console.WriteLine("No, the number I am thinking of is lower than " + userGuess +

                              ". Can you guess what it is?");

        }

    }

    Console.WriteLine("Well done! The answer was " + returnValue);

    Console.WriteLine("Would you like to play again (y/n)?");

    string input = Console.ReadLine();

    playAgain = input.ToLower() == "y";

} while (playAgain == true);

Create a brand new project titled "Guess a Number"

 Random random = new Random();

 int returnValue = random.Next(1, 101);

Create an application that will generate a random number and allow the user to guess the number  
Your application needs at least three methods

 GetUserGuess

Validate that the number is a valid number between 1 and 100

Return the valid integral value

VerifyUserGuess

Takes two parameters

The actual number

The user's guess

Determine if the user guessed correctly

If not, call a third method ProvideAHint that tells the user if the number was higher or lower

Returns whether or not the guess was correct   
Use the value returned from VerifyUserGuess to provide an information message to the user

Use the TryParse method to ensure the user is entering valid numbers