A. Using GITHUB

1) Register an ***email*** address on github with a easy-to-identify ***username***.
2) Go to the Blackboard "test" item (***GITHUB / GITHUB User Info***) on https://blackboard.uhcl.edu/webapps/assessment/take/launchAssessment.jsp?course_id=_19025_1&content_id=_1522218_1
3) to fill out your **github *email*** address and the ***username*** so that we can identify you.
4) For this assignment, go to this link first (the link is also available on Blackboard)— https://classroom.github.com/a/ZgkqilmV
5) Accept the invite and github will create a repository in your account.
6) Using Visual Studio to "clone" the repository on your system: there are some guidelines on Visual Studio with GIT.
   a) For PC, https://docs.microsoft.com/en-us/azure/devops/repos/git/gitquickstart?view=azure-devops&tabs=visual-studio
   b) For Mac, https://docs.microsoft.com/en-us/visualstudio/mac/set-up-git-repository?view=vsmac-2019;
   c) Video Tutorial for Windows: https://channel9.msdn.com/Shows/Visual-Studio-Toolbox/Getting-Started-Faster-with-Git-and-GitHub
7) For each part below, create a solution in Visual Studio within the same cloned repository.
   a) The solution's name should be the same as the filename after the dash (—) symbol.
   b) For example, Part B is called "***Simple App – SimpleApp.sln***". So, the name the solution is "**SimpleApp.sln**"
8) Then open the solution to work on the codes.
9) After you're done, make sure that you've committed your changes and then push these commits to the github server.

B. Simple App – SimpleApp.sln

1) Follow the textbook / PowerPoint slides for **Chapter 3** from **sections 3.2—3.5.**
2) Demonstrate the following.
3) **3.2 Simple App: Displaying a Line of Text**
   a) 3.2.1 Comments
   b) 3.2.2 using Directive
   c) 3.2.3 Blank Lines and Whitespace
   d) 3.2.4 Class Declaration
   e) 3.2.5 Main Method
   f) 3.2.6 Displaying a Line of Text
   g) 3.2.7 Matching Left ({) and Right (}) Braces
4) **3.3 Creating a Simple App in Visual Studio**
   a) 3.3.1 Creating the Console App
   b) 3.3.2 Changing the Name of the App File
   c) 3.3.3 Writing Code and Using **IntelliSense**
   d) 3.3.4 Compiling and Running the App
   e) 3.3.5 Syntax Errors, Error Messages and the **Error List** Window
5) **3.4 Modifying Your Simple C# App**
   a) 3.4.1 Displaying a Single Line of Text with Multiple Statements
   b) 3.4.2 Displaying Multiple Lines of Text with a Single Statement
6) **3.5 String Interpolation**

7) Commit and Push changes to **github** and
8) then verify your changes on **github**.

C. Adding Integers – AddingIntegers.sln

1) Follow the textbook / PowerPoint slides for **Chapter 3** from **sections 3.6—3.7**
2) Demonstrate the following.
3) **3.6 Another C# App: Adding Integers**
   a) 3.6.1 Declaring the int Variable number1
   b) 3.6.2 Declaring Variables number2 and sum
   c) 3.6.3 Prompting the User for Input
   d) 3.6.4 Reading a Value into Variable number1
   e) 3.6.5 Prompting the User for Input and Reading a Value into number2
   f) 3.6.6 Summing number1 and number2
   g) 3.6.7 Displaying the sum with string Interpolation
   h) 3.6.8 Performing Calculations in Output Statements
4) **3.7 Memory Concepts**
5) Commit and Push changes to **github** and
6) then verify your changes on **github**

D. Arithmetic – Arithmetic.sln

1) Follow the textbook / PowerPoint slides for **Chapter 3** from **section 3.8**
2) Demonstrate the following.
3) **3.8 Arithmetic**
   a) 3.8.1 Arithmetic Expressions in StraightLine Form
   b) 3.8.2 Parentheses for Grouping Subexpressions
   c) 3.8.3 Rules of Operator Precedence C# Expressions
   d) 3.8.5 Redundant Parentheses
4) Commit and Push changes to **github** and
5) then verify your changes on **github**

E. Decision Making – DecisionMaking.sln

1) Follow the textbook / PowerPoint slides for **Chapter 3** from **sections 3.9**
2) Demonstrate **3.9 Decision Making: Equality and Relational Operators**.
3) Commit and Push changes to **github** and
4) then verify your changes on **github**

F. Arithmetic 2 – Arithmetic2.sln

Write an app that asks the user to enter two integers and an arithmetic operator (+, -, /, and *) and obtains them from the user. Display their sum, product, difference, and quotient (integer-division) based on the arithmetic operator from the user's input. You need to use an if-statement, arithmetic operators, and write line, read line, and convert. When displaying your results, make sure to make use of string interpolations with $"".

For example,

Enter Number1 => 1

Enter Number 2 => 2

Enter Operator => +

1 + 2 = 3.

G. Integer Equivalent of a Character – IntegerChar.sln

Here's another peek ahead. In this chapter, you have learned about integers and the type `int`. C# also can represent uppercase letters, lowercase letters and a considerable variety of special symbols. Every character has a corresponding integer representation. The set of characters a computer uses and the corresponding integer representations for those characters is called that computer's character set. You can indicate a character value in an app simply by enclosing that character in single quotes, as in `'A'`.

You can determine the integer equivalent of a character by preceding that character with `(int)`, as in `(int) 'A'`.

The keyword `int` in parentheses is known as a cast operator, and the entire expression is called a cast expression. (You'll learn about cast operators in Chapter 5) The following statement outputs a character and its integer equivalent:

```
Console.WriteLine($"The character {'A'} has the value {(int)'A'}");
```

When the preceding statement executes, it displays the character `A` and the value `65` as part of the string. See Appendix C for a list of characters and their integer equivalents.

Using statements similar to the one shown earlier in this exercise, write an app that displays the integer equivalents of some uppercase letters, lowercase letters, digits and special symbols. Display the integer equivalents of the following: `A B C a b c 0 1 2 $ * + /` and the space character.

H. Building a Sentence – Sentence.sln

If you need to be reminded of the syntax, go review the syntaxes on the slides. Create a new console application and follow the following tasks in the Main function:

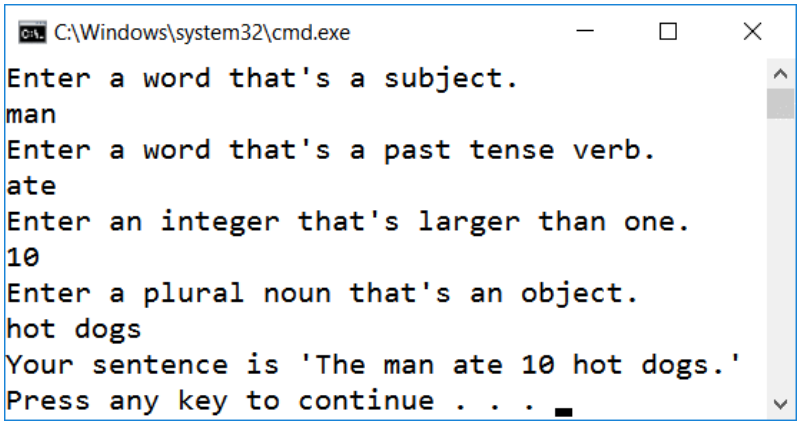
```
static void Main(string[] args)

{

    Console.WriteLine("This app will build a sentence."); // Task 1
```

```
    // Continue to work on Task 2 by declaring the variables, and then complete other tasks.
}
```

1) State that the app will build a sentence.
2) Declare the following variables (four in total):
   a) *number* to be type int, and
   b) *objects*, *subject*, and *verb* to be type string.
3) Prompt the user to enter a subject that's not a pronoun and a proper noun.
4) Read the subject from the user and store it into the variable *subject*.
5) Prompt the user to enter a past tense verb.
6) Read the verb from the user and store it into the variable *verb*.
7) Prompt the user to enter a number that is bigger than one.
8) Read the number from the user and store it into the variable *number*.
9) Prompt the user to enter plural objects.
10) Read the input from the user and store it into the variable *objects*.
11) Now, construct a simple English sentence using the words from the user's inputs: "The {subject} {verb} {number} {objects}."



I. Counting, Negative, Positive, and Zero Value – CountNegPosZero.sln

Create a console application that inputs five numbers and determines and displays the number of negative numbers input, the number of positive numbers input, and the number of zeros input. The code snippet provided will count the number of negatives of the two integer inputs. Please do the same for all five inputs, counting all the positives, negatives, and zeros.

```
static void Main(string[] args)
{
    int negatives = 0;
    Console.WriteLine("Enter the first number.");
    int a = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Enter the second number.");
    int b = Convert.ToInt32(Console.ReadLine());
    if (a < 0)
    {
        negatives = negatives + 1;
    }
    if (b < 0)
    {
        negatives = negatives + 1;
    }
```