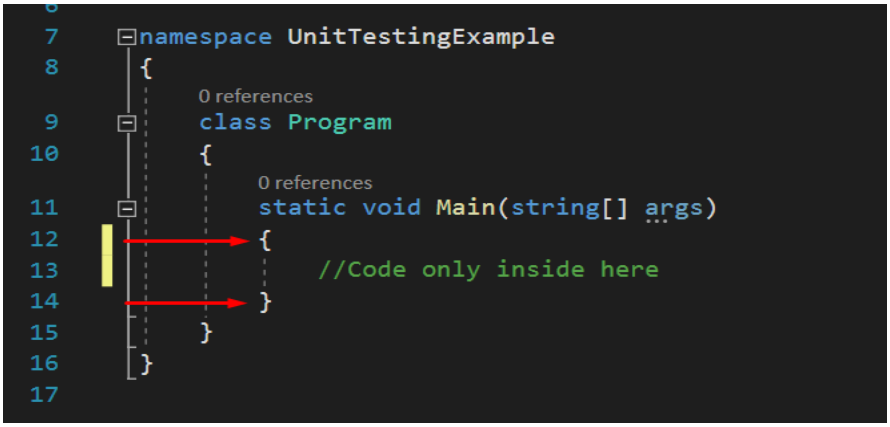
The purpose of this worksheet is to focus down, step by step, on the problem-solving process and offer you a chance to flex your critical thinking muscles. This worksheet will outline the process for breaking down problems into their most simple elements.

You will need to create a console app (.NET framework) in Visual Studio Community 2019. We will be writing all of this code inside the Static void Main.



Don't put any code outside the two curly braces with arrows.

Fun with Variables

1. To start, lets declare an integer called "Value1". Let's give this variable the value of "7".
    a. Let's preform some arithmetic. Let's divide our "Value1" variable by two.
    b. We should capture the result in another variable.
        i. What should the data type be of our "Result" variable?
    c. Let's print the result to the console.
        i. How do we print to the console?
        ii. When you run the code. Does the console open and close immediately? If so, why do you think this is? Is the code running faster than we can see? How can we get the program to "halt execution" so we can see our outputs?
2. Next, let's make some modifications to this logic. We are going need two variables, "Value1" and "Value2". These values are going to be based off user input. We want the user to be able to divide one number by another.
    a. We should be able to capture both values and use them to get a result. Once we have the result, we should print it to the console.
        i. When getting user input what should we always do? How does the user know what to do, what to enter?
        ii. When we capture user input, what data type does it come in? How can we use this to preform mathematical expressions?
        iii. What is something that can go wrong with this logic? Is there a way the user can "break" the application?
3. Let's take a quick pivot to talk about data types.
    a. Given the following, fill in the blanks with the correct datatype. You should declare these inside your code. You will know if it is correct because you will not have errors.
        i. ___ a = 9;

ii. ___ b = 8.99;
iii. ___ c = 'A';
iv. ___ d = false;
v. ___ e = "Hello World";

b. Copy the following code.

i.
```
char h = 'h';
char e = 'e';
char l = 'l';
char o = 'o';
string Greeting = h + e + l + l + o;
Console.WriteLine(Greeting);
Console.ReadLine();
```

ii. We want this code to print "hello" to the console. Without changing the variable data type declarations (I.e "char h = 'h') how can we get this to build into a string?
iii. What does the error say? Why would it consider our characters as integers?
iv. Valuable research terms:
    1. "why does char + char give integer in c#"

<u>Fun with For-Loops</u>

1. Write a for loop that will run 5 times.
2. Modify the loop to print out every incrementation of the for-loop's counter. (Usually "i")
    a. I.e 0,1,2,3,4
3. Using the same loop, change it to now print backwards from 9.
    a. 4,3,2,1,0
4. Write a loop that will run as many times as a user wants.
    a. Change the loop to run "Forward" again.
    b. Take in user input and use the value to run the for loop.
    c. Valuable research terms:
        i. "How do I capture user input in C#"
        ii. "How do I convert a string into an int in C#"
5. We are now going to modify this loop to run as many times as there is letters in a string.
    a. Declare a string above the for loop with a value of "Hello World".
    b. Your loop should run 11 times for each letter in the string.
        i. Note: using " i < 11" is not what we are looking for. I should be based off the string itself. So that we could have this working for any string, regardless of how many characters is present.
    c. Your loop should now output: 0,1,2,3,4,5,6,7,8,9,10
    d. Valuable research terms:
        i. "How do I get the number of characters in a string C#"
6. Using the same loop from step 4. Let's modify the logic to instead print out every letter of our string one at a time.
    a. I.e "H,e,l,l,o, ,W,o,r,l,d"

   b. Valuable research terms:
      i. “How to print one letter from a string”
7. Using the same loop from step 6. Modify it to print out every OTHER letter.
   a. I.e “H,l,o,W,r,d”
8. Let's take this same loop and modify it to only print out if the current index is divisible by 3.
   a. I.e “H,l,W,l”
   b. Valuable research terms:
      i. “How to find if a value is PERFECTLY divisible by another number”

Fun with While-Loops

1. Write the following while loop.
   a.
```
int counter = 5;
while(counter < 25)
{
    counter++;
}
```
   b. Questions to think about:
      i. How many times will this loop run?
      ii. What is the difference between this while loop and a for loop?
2. Let's now modify this loop to take in user input for our counter instead of hard coding a value. Our user should be able to enter any number they choose to start off our counter variable.
   a. You are going to need to capture the input from the user then turn it into an integer.
   b. Questions to think about:
      i. What is something you think could go wrong with this set up?
      ii. How do you think we could fix this problem?
      iii. How many possible times do you think this loop will iterate?
      iv. What if they enter a negative number?
3. Let's continue to tweak this loop. Let's change the condition for the loop to check to see if our counter is not equal to 42. If the value is not equal to 42 then we should re-prompt the user until they enter the correct number. Just for clarification for the user; we should add some outputs to our logic. We should both ask our user to input the initial value with something like “Please enter the number 42”. And another output for if the user enters it wrong.
   a. Questions to think about:
      i. How many times is this loop guaranteed to run?
      ii. Is there potentially another type of while loop we could use for this logic?