Function, Loop, and String Exercise: Table of Contents Display

Read the entire instructions before starting. As in assignment 5, submit snapshots of your code for all 5 steps (but not helper steps). This assignment is about using variables and passing parameters. Only the code in main() knows book details like title, chapter number, and page number. All the other functions do not assume any particular title, chapter number, or page number. Those values are passed as inputs to produce the table of contents for any book.

In order to use both cout and the string variable type, your cpp file will need lines at the top for both `#include <iostream>` and `#include <string>.`

The overall project is to create an app to print the table of contents for a book. Create a `main()` function with 5 string variables, three of which are called sCh1Name, sCh2Name and sCh3Name and two which are called sSec3_1Name and sSec3_2Name. Use the chapter names: "Introduction", "Using Variables", and "Using If Statements". The sections in the last chapter (chapter 3) are "If" and "Else". Here is an example:

```
string sCh1Name = "Introduction";
```

Step one:
Create two separate functions, one to print a single chapter number and title, and the other to print a single section number and title. The functions will take the number as an int parameter and title as a string parameter. When the chapter function is called with the number 3 and the title string "Using If Statements" it should print the following line to cout:

```
Chapter 3: Using If Statements
```

The section function, when called with the number 2 and the string "Else" should print to cout:

```
  Section 2: Else
```

Note that the word "Section" is preceded by two spaces.
Call the printChapterLine function from your main three times, and printSectionLine twice:

```
printChapterLine(1, sCh1Name);
printChapterLine(2, sCh2Name);
printChapterLine(3, sCh3Name);
printSectionLine(1, sSec3_1Name);
printSectionLine(2, sSec3_2Name);
```

Step two:
Create a function that prints a series of dots using the period character. It will take a count and print that many dots to cout, using a loop. Look at the for loop on page 264 of the Gaddis, 7th Edition book. Your dots function takes an int parameter for the length. Use that parameter in the for loop's test expression, instead of the 5 that is used by the book to loop 5 times. To test your function, call it directly from main, e.g. `dots(7);`

Step three_helper:
In two of the steps below, you need to know how many characters are needed to print the value of an integer variable for chapter number and for page number. Create a function that takes an int as a parameter and returns the number of characters in the printed representation of the input

value. Here is a clue about how to determine the number of characters needed for an integer: >9999 needs 5, >999 needs 4, >99 needs 3, >9 needs 2, else 1

Step three:
Modify the first two functions so they return the length of the line they just sent to cout, and to not put an endline at the end of what they print. (It may also add one space at the end. Look at the example below.) To determine the length of what was printed, you will have to add up the lengths of what you print. For parts that are of type string, use the following information.
The string type is a Class, as we shall learn later. You can use its size function to get the length:

```
int iTitleLength = sTitle.size();
```

Step four:
Combine the functions in steps two and three to print the chapter or section name, figure out how many spaces are left in a 62 column line, and fill the remaining space in the line with dots. The number of dots you need to print is 62 – (length so far).

Step four_helper:
Create a helper function that fills a line out to the specified length with 1234567890 repeating (see example below). Use the helper line to verify that you have lined things up correctly – it can save you from having to count letters and dots. The dotCountHelper function can be written with a single loop and just the loop counter variable. The count must start with one and end with 62. The value that you print is the loop counter modulo 10 (the remainder that you get if you divide the number by 10).

```
cout << i % 10;
```

Step five:
Add to step four by putting the page number at the end of the line. Use the function from step three_helper to figure out how many characters are needed for the page number. You will have to modify your main and possibly other functions to store or pass the page numbers as well as the chapter names and numbers.

The final result after step 4 should look like:

```
Chapter 1: Introduction .....................................
Chapter 2: Using Variables ..................................
Chapter 3: Using If Statements ..............................
  Section 1: If .............................................
  Section 2: Else ...........................................
12345678901234567890123456789012345678901234567890123456789012
```

The final result after step 5 should look like:

```
Chapter 1: Introduction ..................................... 1
Chapter 2: Using Variables ................................. 18
Chapter 3: Using If Statements ............................. 29
  Section 1: If ............................................. 30
  Section 2: Else ........................................... 37
12345678901234567890123456789012345678901234567890123456789012
```