Computational Problem Solving (CS100)

Shafay Shamail, Arif Zaman and Safee Ullah Chaudhary

Lab 7 Objectives:

- Practice strings and nested loops.

Lab Guidelines

1. Make sure you get your work graded before the lab time ends.
2. Upload all your work onto the LMS folder designated for the lab (i.e. “Lab07”) before the time the lab ends.
3. Talking to each other is NOT permitted. If you have a question, ask the lab assistants.
4. If you are a hot-shot C++ expert, you are still not allowed to use any feature of C++ that has not been covered in class or in the lab.
5. The objective is not simply to get the job done, but to get it done in the way that is asked for in the lab.

Task 1: Cheat Sheet

You are to create a cheat sheet for grade 5 students so they have better access to multiplication tables.

Using a loop, generate the multiplication table of 6. It should look something like this,

```
6 12 18 24 30 36 42 48 54
```

Now modify the code to generate the multiplication table of 4, like this

```
4  8 12 16 20 24 28 32 36
```

Further edit the loop to generate a multiplication table of any integer that the user enters

STOP AND SHOW YOUR CODE TO THE TA

Now using your previous code and nested loops, generate the following multiplication table.

```
1  2  3  4  5  6  7  8  9
2  4  6  8 10 12 14 16 18
3  6  9 12 15 18 21 24 27
4  8 12 16 20 24 28 32 36
5 10 15 20 25 30 35 40 45
6 12 18 24 30 36 42 48 54
7 14 21 28 35 42 49 56 63
8 16 24 32 40 48 56 64 72
9 18 27 36 45 54 63 72 81
```

STRINGS

- Strings are sequences of characters.
- To use the strings, you have to include the string library by typing #include <string>
- The + operator is used to *concatenate* two or more strings or strings and characters;

  put them together to yield a longer string.
- The length() function of strings can be used to yield the number of characters in a string.
- The dot notation is used to call functions which are related to strings. So, for example, if you want to find the length of a string, you write the string whose length you want, then a period, then the name of the function, followed by parentheses:

```
string mystring = "Hello World";
int str_length = mystring.length();
cout << "Length of my string: " << str_length;
```

Output of the above code will be:

```
Length of my string: 11
```

Notice how the dot notation is used to call the length function

- The substr(A,B) function is used to extract sub strings from within a string. So, for example, if you want to get "ello Wo" from the string "Hello World", you can use the substr(A,B) function.
- The substr(A,B) function starts counting from the Ath position and takes the next B-1 characters.
- In strings, the index of the first character is 0. So if you want to extract "Hello W" from the string "Hello World",

H	**e**	**l**	**l**	**o**		**W**	**o**	**r**	**l**	**d**
0	1	2	3	4	5	6	7	8	9	10

The starting index (value of A) will be 0 and B will be 7 as you want the first 7 characters.

```
string mystring = "Hello World";

string sub_string = mystring.substr(0,7);

cout << "substring: " << sub_string;
```

Output of the above code will be:

```
substring: Hello W
```

Similarly, if you want to extract "ello Worl" , The starting index (value of A) will be 1 and B will be 7 as you want the first 9 character.

Task 2: Alphabet Counter

In a given string, the number of spaces can be counted using the following code:

```
string str = "Mary had a little lamb";
int counter = 0;

for(int i=0; i<str.length();i++)
{
        if(str.substr(i,1) == " ")
                counter++;
}
cout << "Number of spaces: " <<  counter << endl;
```

The output of the above code will be:

```
Number of spaces: 4
```

Note: *substr(A,B) function outputs a string and not a character.*

You can use this code and test its working until you're comfortable with the way it works. Modify the above code to create a program which will take as input any string that a user enters and returns the frequency of each alphabet present in it.

HINT:

```
char mychar = 'a';
mychar = mychar + 1;
cout << "new character: " << mychar << endl;
```

The output of the above code will be:

```
new character: b
```

STOP AND SHOW YOUR CODE TO THE TA

Task 3: Number Triangle

Nested loops can create amazing patterns if executed carefully.

In this task you will use loops to create the patterns shown below. Your program should take the height (in red) of triangle as input and output the corresponding pattern as shown below.

HINT: Note the number of spaces before each line starts.

HINT: A row (in blue) of this pattern starts from a number 'x' (4 in this case) and ends at 'x'. You may first want to write a code using loops which prints a line from 'x' to 1 and back to 'x'. Then use this code along with the code for adding appropriate spaces to generate the pattern.

N=6	N=5	N=4	N=3
1 212 32123 4321234 543212345 65432123456	1 212 32123 4321234 543212345	1 212 32123 4321234	1 212 32123

STOP AND SHOW YOUR CODE TO THE TA

b) For this part, modify the code from the previous problem such that it outputs all the patterns from 2 to N. So if the user inputs N=5, the output of your code should be patterns for N=2, N=3, N=4 and N=5.

STOP AND SHOW YOUR CODE TO THE TA