Programming lab 3.

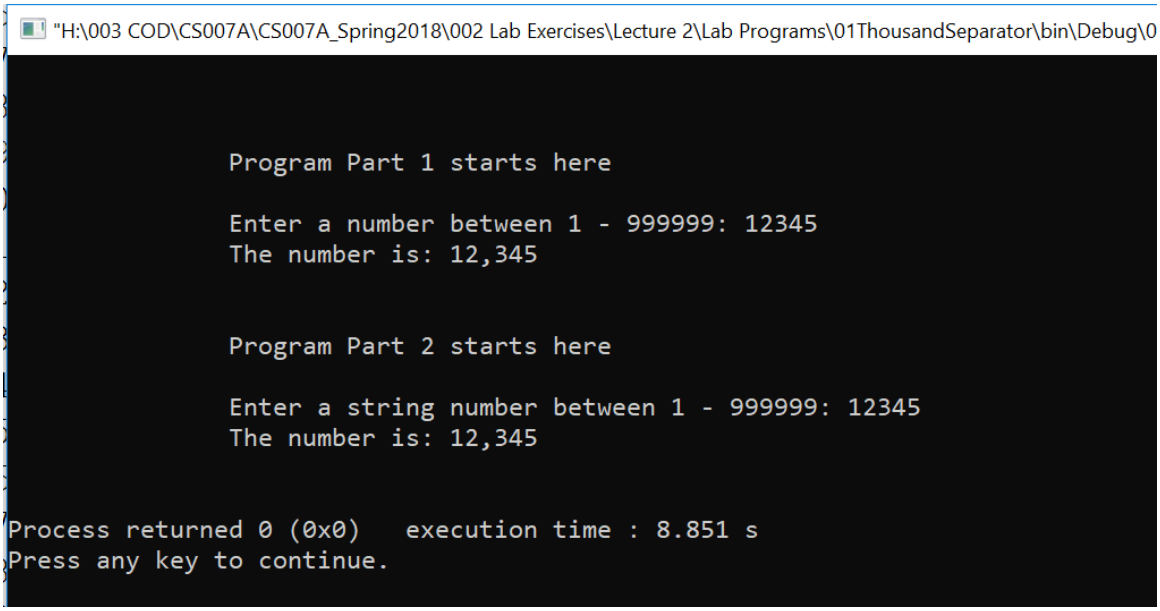
Exercise 1. ThousandSeperator

This program has two parts to it. In the first part you are going to enter a number between 1 – 999999 from the keyboard and store that number in an integer variable. Now you are going to insert a thousand separator. E.g. if you enter 1000 you are going to print out 1,000. I urge you to try the following numbers. 500, 1000, 1234, 10000, 12345, 100000, 999999

In the second part of this program you are going to read a number as a string that has the same range. Now insert a 1000 separator to the number. Try the same numbers as above to confirm that your program works.

See printout below what you are going to print out to the console window.

Name the cpp file for this exercise: **01ThousandSeparator.cpp**



Exercise 2. Number or Character

This exercise give you a chance to practice decision making. The program is going to generate a number or a character. If a number is generated (between 1 – 100), we need to decide if the number is even or odd. The range of characters goes from colon (:) to tilde (~). Use the ASCII table to figure out the range for those characters.

The program has a timing block that uses the sleep( ) function. The function takes the time in seconds and we are going to use sleep(1) for one seconds. A star should pop up every second until 5 stars has been printed. Take notes during the demo.

The program shall have three line shift top margin and 10 spaces left margin. Beyond that, be creative in your execution with regards on how you want to show the result.

Writing pseudo code (recipe) on how you want to solve this could be smart. I have used three random generators to solve this. You cannot do it with less. Also you might benefit from using nested if statements.
Finally, make sure you test your program to see that the program generated both even and odd numbers as well as a character in the described range.

Name the cpp file for this exercise: **02NumberOrCharacter.cpp**

Exercise 3. ColorMixer

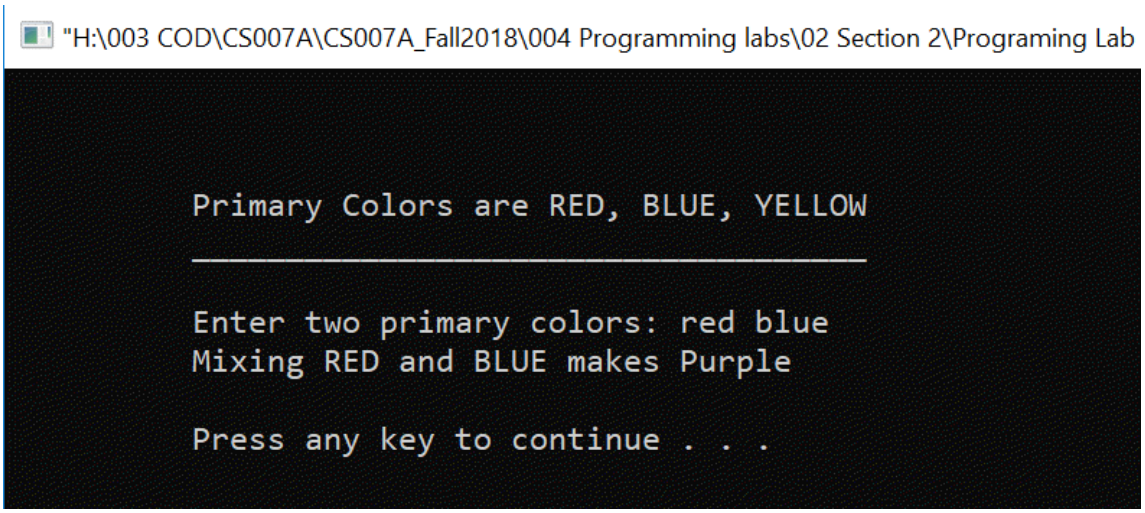
The colors red, blue and yellow are known as the primary colors because they cannot be made by mixing other colors. When you mix two primary colors you get a secondary color as shown here:

When you mix red and blue, you get purple.
When you mix red and yellow, you get orange.
When you mix blue and yellow you get green.

Write a program that prompts the user to enter the name of two primary colors to mix. If the user enters anything other than “red,” “blue,” or “yellow,” the program should display an error message. Otherwise, the program should display the name of the secondary color that result.

Tips: Use trailing else to obtain error check. Also you might be better off to convert your entry to uppercase. You do that by including algorithm. The use the following statement.
transform(color1.begin(), color1.end(), color1.begin(), ::toupper);

Name the cpp file for this exercise: **03ColorMixer.cpp**

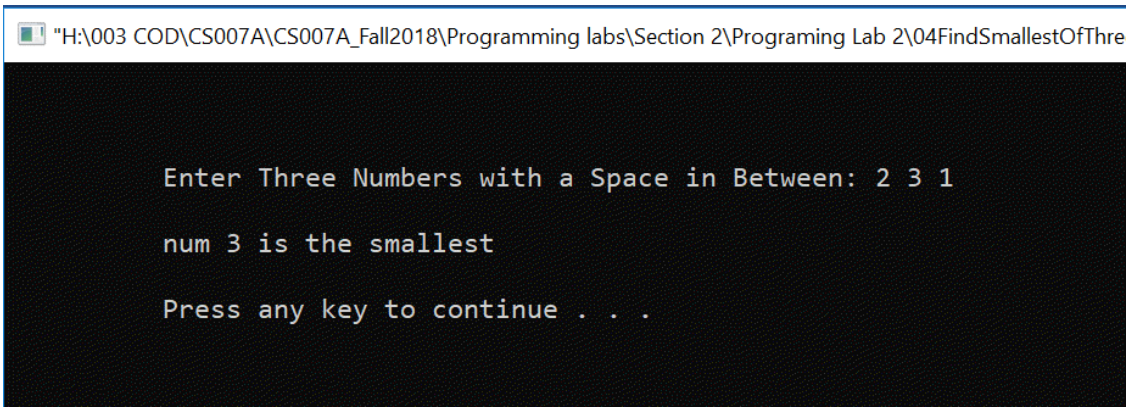


Exercise 4. Find Smallest of Three Numbers

In this exercise you are going to read three integers from the keyboard and the program will print out which is the smallest of the three numbers. Try all possible combinations to make sure that your algorithm is correct. E.g. 123, 132, 213, 231, 312, 321.

To accomplish this, use nested if statements. This could be achieved with combined statements in the if's but use nested if which means one condition in each if statement

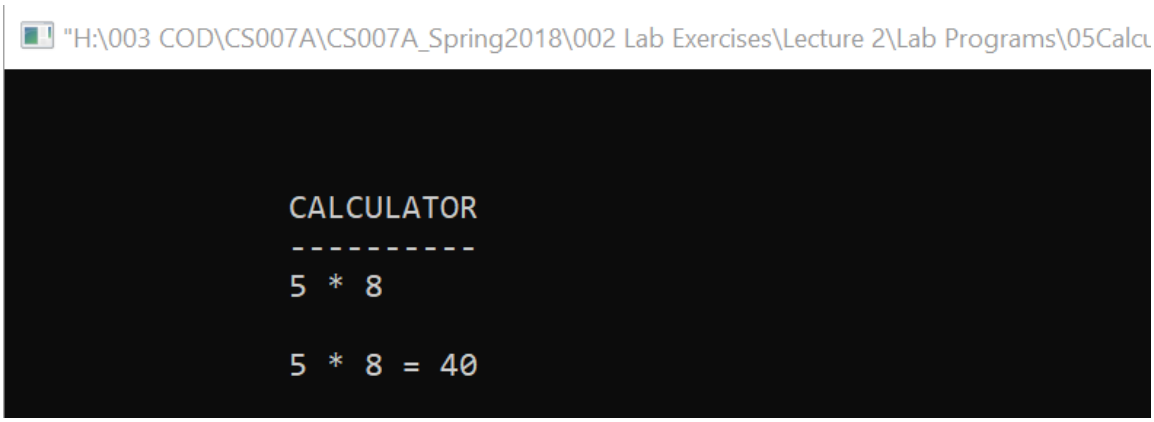
Name the cpp file for this exercise: **04FindSmallestOfThree.cpp**

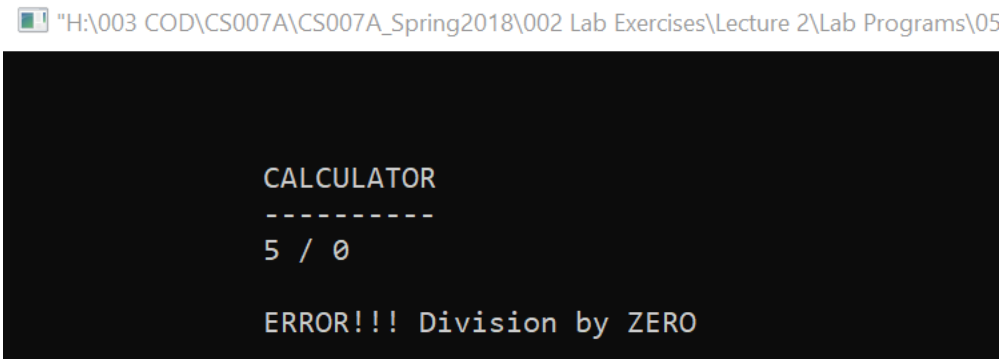


Exercise 5. Calculator

Write a program that mimics a calculator. The program should take two integers and the operation to be performed. It should then output the numbers, the operator and the result. (For division, if the denominator is zero, output an appropriate message.) See picture below for multiplication. Make sure you try your program for addition, subtraction, multiplication, division and modulus. Implement the logic using SWITCH block rather than if statements.

Name the cpp file for this exercise: **05Calculator.cpp**





Exercise 6. ParkingFee

The short-term parking fee, 0 – 24 hours, parking fee, F at an international airport is given by the following formula

$$F = \begin{cases} 5, & if\ 0 \le h \le 3 \\ 6\ X\ int(h+1), & if\ 3 < h \le 9 \\ 60, & if\ 9 < h \le 24 \end{cases}$$

Where *int(h + 1)* is the integer value of *h + 1*. For example, *int(3.2) = 3*, *int(4.8) = 4*. Write a program that prompts the user to enter the number of hours a car is parked at the airport and outputs the parking fee.