CS-181 Lab 4

Student Learning Outcomes

- More experience with the Microsoft Visual Studio Integrated Development Environment (IDE)
- More experience Creating C++ projects in MS Visual Studio
- More experience using the code editor
- More experience writing sequential code
- Conditional expressions
- Arithmetic expressions
- Formatting output

Overview

This lab provides a hands-on opportunity to apply some of the terminology and concepts presented in chapter 4.

Exercise 1: Roman Numeral Converter

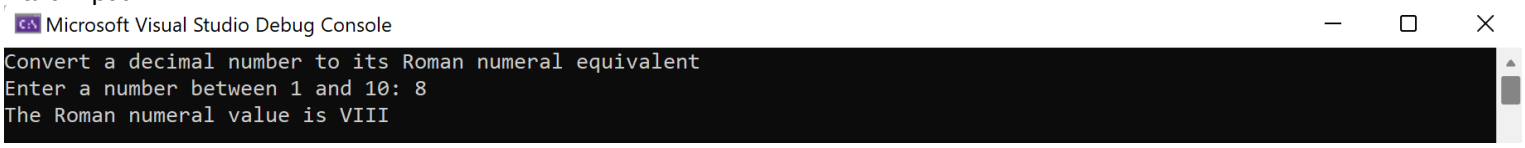
Write a program that prompts the user to enter a number between 1 and 10 and display the equivalent Roman numeral. Hint: a switch statement will make this easier. You must have input validation and not allow a value less than 1 or greater than 10. Be sure to test your program with both valid and invalid numbers.

Conversion table

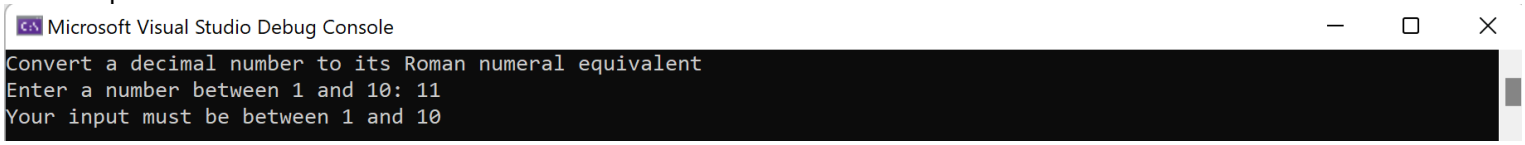
Number	Roman Numeral Value
1	I
2	II
3	III
4	IV
5	V
6	VI
7	VII
8	VIII
9	IX
10	X

While there is some room to be creative, user prompts, error messages and output need to be clear. Your program output should look something like this:

Valid input:



Invalid input:



Exercise 2: Body Mass Index

Write a program that calculates and displays a person's body mass index (BMI). The BMI is often used to determine whether a person is overweight or underweight for his or her height. A person's BMI is calculated with the following formula:

$BMI = weight * 703/height^2$

where weight is measured in pounds and height is measured in inches. The program should display a message based on the following ranges:

- Underweight: BMI is less than 18.5
- Normal weight: BMI is 18.5 to 24.9
- Overweight: BMI is 25 to 29.9
- Obese: BMI is 30 or more

Format your BMI output to 2 decimal places.

Your program output should look something like this:

Exercise 3: Software Sales

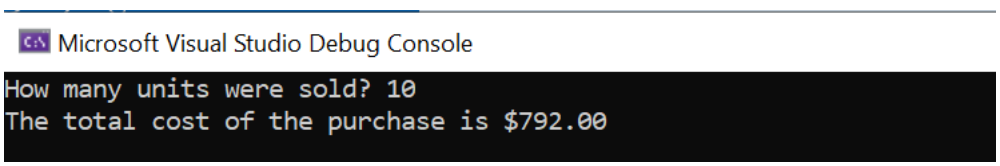
A software company sells a package that retails for $99. Quantity discounts are given according to the following table.

Quantity	Discount
10–19	20%
20–49	30%
50–99	40%
100 or more	50%

Write a program that asks for the number of units sold and computes the total cost of the purchase.
Input Validation: Make sure the number of units is greater than 0.

Format your output to 2 decimal places.

Your output should look something like this:



Be sure to test multiple ranges.

Exercise 4: Shipping Charges

The Fast Freight Shipping Company charges the following rates:

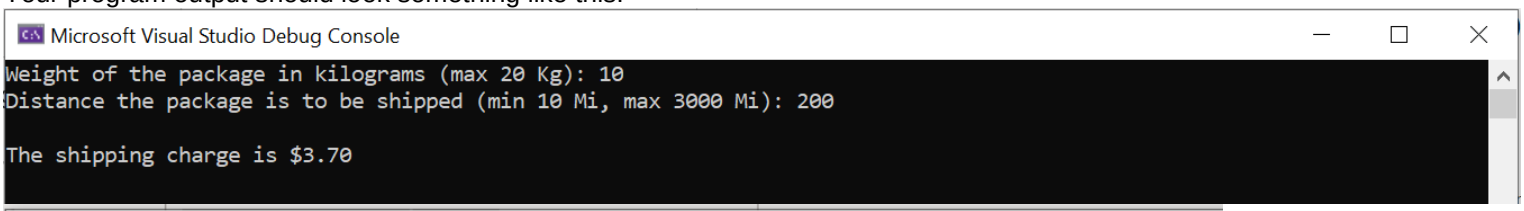
Weight of Package (in Kilograms)	Rate per 500 Miles Shipped
2 kg or less	$1.10
Over 2 kg but not more than 6 kg	$2.20
Over 6 kg but not more than 10 kg	$3.70
Over 10 kg but not more than 20 kg	$4.80

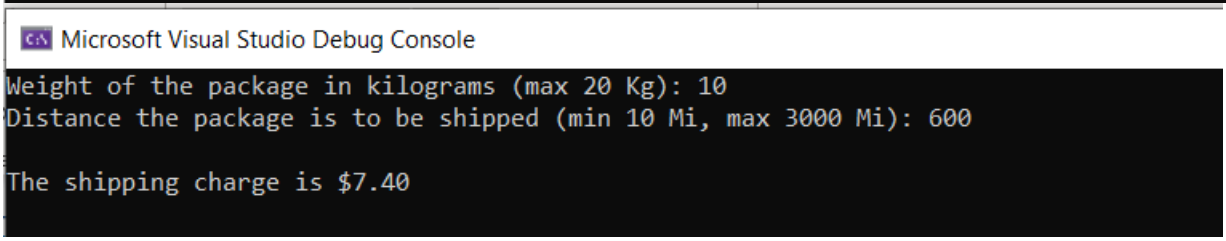
Write a program that asks for the weight of the package and the distance it is to be shipped, then displays the charges.

Input Validation: Do not accept values of 0 or less for the weight of the package. Do not accept weights of more than 20 kg (this is the maximum weight the company will ship). Do not accept distances of less than 10 miles or more than 3,000 miles. These are the company's minimum and maximum shipping distances.

Format your output to 2 decimal places.

Your program output should look something like this.





Be sure to test multiple ranges.

Hint: Dealing with mileage ranges can be a bit tricky. Here's a formula that will work. Be sure to adjust the rate for the package weight: shippingCharge = 1.10 + (int((distance - 1) / 500) * 1.10);

Assignment Submission

Your completed assignment must be uploaded to Canvas in a zip file format. Be sure you have folders for all 4 exercises in your Lab04 folder and you have all files and sub-folders needed to open with Visual Studio.

Grading Criteria:

Deliverable	**Points**	**Breakdown**
Exercise 1	10	Opens in VS, code is clear (I.e., appropriate variable names, use of constants, comments, program header), compiles, runs, produces correct output. Output formatted correctly
Exercise 2	12	Opens in VS, code is clear (I.e., appropriate variable names, use of constants, comments, program header), compiles, runs, produces correct output. Output formatted correctly
Exercise 3	14	Opens in VS, code is clear (I.e., appropriate variable names, use of constants, comments, program header), compiles, runs, produces correct output. Output formatted correctly
Exercise 4	14	Opens in VS, code is clear (I.e., appropriate variable names, use of constants, comments, program header), compiles, runs, produces correct output. Output formatted correctly
Lab Total	**50**	