CPSC 1155 – Project 1

Sequential Programming

Objectives

The goal of this assignment is to develop sequential pseudocodes, algorithms and C++ programs using math functions, selections, and random numbers.

Readings

You should be reading Chapters 1, 2, and 3 of the textbook. The lectures and labs will provide additional supporting material.

Instructions

For each of the following problem statements:

1. Read the problem statement and clarify the problem.
2. Develop the algorithm that solves the problem and determine input, output, and processing (IPO) along with the necessary variables. Use meaningful variable names.
3. Write a **Pseudocode**, as required, and a **C++ program** that implements the algorithm. Add extra comments where needed.

Make sure you use a comments header to reflect the intention of your program and name of the author (you) and the date the program was written.

1. Test, debug, and execute the program using typical values.
2. Submit according to the instruction in the "Lab Submissions" section.

Problem Statements

1. (volume\_sphere.cpp) Write a **C++ program** that calculates and displays the volume of a sphere. The program should prompt the user to enter the radius of the sphere. Assume the input is positive. Here is a sample run:

Enter radius of sphere: 1.5

The volume of sphere is 14.14

Make sure to display the output with 2 decimal places. Please check how to use [M\_PI](https://docs.microsoft.com/en-us/cpp/c-runtime-library/math-constants?view=vs-2019) and [pow](http://www.cplusplus.com/reference/cmath/).

1. (swimming\_pool.cpp)[[1]](#endnote-1) Write a **pseudocode** and a **C++ program** that solves the following problem. Two large and 1 small pump can fill a swimming pool in 4 hours. One large and 3 small pumps can also fill the same swimming pool in 4 hours. The user will enter how many large pumps and small pumps are used. Display how many hours it will take to fill the swimming pool with the specified numbers of pumps. Assume user inputs are larger or equal to 0 and are within reasonable ranges. Here is a sample run:

Enter the number of large pumps:4

Enter the number of small pumps:4

It will take 1.66667 hours to fill up the pool with 4 large pumps and 4 small pumps.

Hint: This problem requires you to find the rate of small pump and rate of large pump first. Then we can calculate the hours. See end note i for the original problem and starting equations. You will need to deduct the equations for finding the rate yourself first before programming this problem.

1. (wind\_chill\_temperature.cpp, exercise 2.17 in the text) How cold is outside? The temperature alone is not enough to provide the answer. Other factors including wind speed, relative humidity, and sunshine play important roles in determining coldness outside. In 2001, the National Weather Service (NWS) implemented the new wind-chill temperature to measure the coldness using temperature and wind speed. The formula is:

where is the outside temperature measured in degrees Fahrenheit and  is the speed measured in miles per hour. is the wind-chill temperature. The formula cannot be used for wind speeds below 2 mph or temperatures below –58°F or above 41°F (For now you do not have to test this!).

Write a **pseudocode** and a **C++ program** that prompts the user to enter a temperature between –58°F and 41°F and a wind speed greater than or equal to 2 and displays the wind-chill temperature. Use pow(a, b) to compute .

Here is a sample run:

Enter the temperature in Fahrenheit (must be between -58°F and 41°F): 5.3

Enter the wind speed miles per hour (must be greater than or equal to 2): 6

The wind chill index is -5.56707

1. (complex\_number.cpp) We can multiply two complex numbers by using the following formula:

Where is the first complex number and is the second imaginary number. We can simplify the formula by multiplying the terms and simplifying them.

Write a **C++ program** that prompts the user to enter two complex numbers (the first number is *x* and the second number is *y*) and calculates the product of them.

Hint: . See [here](https://www2.clarku.edu/faculty/djoyce/complex/mult.html) for an example and formula.

Here is a sample run:

Enter the first complex number (real number followed by imaginary number): 3 2

Enter the second complex number (real number followed by imaginary number): 1 4

The product of 3 + 2i and 1 + 4i is -5 + 14i

The inputs can have decimal numbers. Check [showpos](http://www.cplusplus.com/reference/ios/showpos/) and [noshowpos](http://www.cplusplus.com/reference/ios/noshowpos/) to help you print the + and – sign.

1. (is\_right\_side\_triangle.cpp) Write a pseudocode and C++ program that reads three positive integers for the edges of a triangle and determines if it is a right triangle. If it is, the program displays that the triangle is a right triangle. Otherwise, the program displays that the triangle is not a right triangle. The program does not accept negative values.

Hint: A triangle is a right triangle if one angle is 90° degrees or radian (What is cos (90°)?). We can use the law of cosines to find the angles. Note that the order of a, b, and c is important in the formula.

Here are two sample runs:

Please enter lengths of a triangle: 3 4 5

The triangle with lengths 3 4 5 is a right-side triangle

Please enter lengths of a triangle: 4 5 6

The triangle with lengths 4 5 6 is a not a right-side triangle

Please enter lengths of a triangle: -3 4 5

The values must be positive

1. (simple\_caculator.cpp) Write a C++ program that asks the user to enter two integers and an arithmetic operator (+, -, \*, /). Then based on the operator, the program calculates the result. The program checks input for invalid operators. It also checks for division by 0.

Here are sample runs:

Enter two integers: 4 8

Enter an arithmetic operator (+, -, \*, /): +

Result = 4 + 8 = 12

Enter two integers: 4 8

Enter an arithmetic operator (+, -, \*, /): $

Invalid operator.

Enter two integers: 4 0

Enter an arithmetic operator (+, -, \*, /): /

Cannot divide by 0.

1. (sort\_three\_integers.cpp ) Write a C++ program that prompts the user to enter three integers and displays the integers in decreasing order.

Requirement: Compare every two integers and swap them if needed. Three swaps might be needed.

Here is a sample run:

Enter three integers: 1 2 3

Sorted in descending order: 3 2 1

1. (pick\_card.cpp, exercise 3.21 in the text) Write a C++ program that simulates picking a card from a deck of 52 cards. Your program should display the rank (Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King) and suit (Clubs, Diamonds, Hearts, Spades) of the card. Here is a sample run:

The card you played is Jack of Hearts

Project Submissions

Submit a zip folder named as yourName\_Project1.zip to Brightspace. This folder should consist of the **C++ codes** in individual .cpp files and one pseudocode.txt file with all your **pseudocode** (for question 2, 3, and 5). See the document about coding styles on Brightspace.

Please make sure that all your .cpp files compile and run properly before submission. Your file must run properly in order to receive full marks.

Marking Scheme

There are 10 marks for each question, total 80, with the following details:

Question 1: volume\_sphere.cpp

* 5 for correct solution
* 4 for usage of pow and M\_PI
* 1 for output with 2 decimal places

Question 2: swimming\_pool.cpp

* 5 for pseudocode
* 5 for correct solution

Question 3: wind\_chill\_temperature.cpp

* 5 for pseudocode
* 5 for correct solution

Question 4: complex\_number.cpp

* 8 for correct solution
* 2 for output with correct +/- signs using showpos and noshowpos

Question 5: is\_right\_side\_triangle.cpp

* 5 for pseudocode
* 5 for correct solution

Question 6: simple\_calculator.cpp

* 2 each for calculating typical + - \* / cases
* 1 for checking invalid operator
* 1 for checking division by 0

Question 7: sort\_three\_integers.cpp

* Must use swap

Question 8: pick\_card.cpp

* 10 for correct solution

1. Adapted from Q1 in [this math problems set](https://www.analyzemath.com/high_school_math/grade_12/problems.html) [↑](#endnote-ref-1)