COSC 1435 Introduction to Problem Solving with Computers I

Homework Assignment 4

* Introduction to C++
* Algorithms (Decision Structure)

1. Briefly describe the difference(s) between a *character literal* and a *string literal*. How is each stored in memory? [4 points]

A character literal can only store one value like A, 4, or ! stored in 8bits or 1 byte

A string literal can store many characters “Hi” “My name is Kevin” , or “how are you” stored in 4 bytes

1. Briefly describe what happens when a floating-point **literal** is assigned to an integer-type **variable**. [4 points]

When a floating-point literal is assigned to and integer everything after the decimal is ignored. Example: int x = 1.95

x = 1

1. Briefly describe what *integer division* is and under what circumstances it will occur. [4 points]

Integer division is when both numbers are integers and if they do perfectly divide the decimal part of the number will be ignored

1. Below is a short C++ program that contains some syntax errors. Find and highlight the errors in yellow. Briefly, in short sentences, describe why each is an error and what should be the correct statement. *Refer to the line numbers when describing the errors.* [8 points]

**Line #**

1 include <iostream>

2 using namespace std;

3 int main()

4 }

5 float gallons,

6 #milesDriven,

7 avg;

8 cout >> "How much gas can your car hold?;

9 cin << gallons;

10 cout << "How far did you drive?";

11 cin >> #milesDriven;

12 #milesDriven / gallons = avg;

13 cout << "Your average MPG is", avg;

14 return 0;

1. {

On line 1 we need #include instead of include. On line 6 you can’t start a variable name with a #. On line 8 the arrows for cout are the wrong way, and there is no closing quotation marks. On line 9 the cin arrows are the wrong way. On line 12 the left type variable is on the right side and vice versa.

1. What would each of the following arithmetic expressions evaluate to **in C++?** [5 points]

|  |  |
| --- | --- |
| **Expression** | **Value** |
| 7 + 2 \* 3 | 13 |
| 4 + 19 % 4 – 1 | 6 |
| 8 + 10 / (6 – 3) | 11 |
| (4 + 3) \* 2 | 14 |
| (7 – 2) \* (4 + 6 / 2) | 35 |

1. Below is a short C++ program. Write the missing “#include” statement and the missing C++ statement that will display results to 2 decimal places. *The 2 decimal places should be displayed even if there are only trailing zeroes*. [5 points]

#include <iostream>

#include<iomanip> **//**🡨**Missing #include**

using namespace std;

int main()

{

float cost, tax, total;

cout << "Please enter the cost of the item: ";

cin >> cost;

cout << "Please enter the tax rate: ";

cin >> tax;

total = cost \* (1 + tax);

**//Write statement for formatting on next line**

cout << setprecision(2) << fixed;

cout << "The total cost will be $" << total << "\n";

return 0;

}

1. Design an algorithm and represent it in **pseudocode** **or flowchart** to determine whether any 5-letter word is a palindrome. Palindromes are words or sentences that read the same backward or forward. For example, “kayak” is a palindrome while “meter” is not. Ask the user to input **5 characters**. You will need five separate variables to store these five characters. After obtaining the characters, compare the characters to determine if the word is a palindrome and output an appropriate message. [20 points]

Print “enter 5 letters with a space between each character”

Get a, b, c, d, e

If((a == e) && (b == d))

Print “Your word is a palindrome!”

Else

Print “Your word is not a palindrome…”

End if

End

1. Write a program, in C++, for a catering company that charges $10 per adult and $6 per child. The program asks the user to enter the number of adults and the number of children to attend the party. Compute the amount to charge then add 8.25% sales tax before displaying the total. The name of the submitted program must be **catering.cpp** [20 points]
2. Write a program, in C++, that gets a five-digit number from the user, separates the number into its individual digits and prints the digits separated from one another by three spaces each. For example, if the user enters 82339, the program should print what is shown in the sample output. You will use integer division and the modulus operator to break down the number. The name of the program must be **separate.cpp** [30 points]

**Sample Run**

Please enter a five-digit integer:**82339 ↵**

The five digits are:

8 2 3 3 9

*Programming/logic hints:*

1. Use a **single** int variable to read the input from the user.
2. cout the number divided by 10000 to get the leftmost digit. *Why does this work?* *Hint: Refer to your answer for integer division.*
3. Change the number to a 4-digit number using the modulus operator. i.e. num % 10000 evaluates to the integer remainder—in this case, the right-most four digits.
4. Repeat the logic (steps 2&3) to display the five digits

**Instructions and what to submit**

* For programming questions, make sure that you follow a good programming style. In each program, include a header comment with the following information:

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Name:

//Class: COSC 1435 Fall 20

//Instructor: Marwa Hassan

//Assignment x Problem x

//Date:

//Program description: **WRITE what this program actually does**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* Place the files of the homework assignment in the folder firstInitialLastNameAssignX, where X is the homework number. Example, for the name Joe Smith, the folder name must be **jsmithAssignX**.
* The submitted **zipped** folder should contain:

1. **Assign4.docx** (for questions **1 - 7,** copy the question and clearly list the answer under it. Please **include your name at the top** of the document)
2. **catering.cpp & separate.cpp (do not submit output files please)**

**Grading criteria for submitted code**

**Programs are graded separately out of 100. *Final assignment grade is weighted*.**

|  |  |
| --- | --- |
|  | Grading |
| Program correctness (Program outputs the correct answer) | 70 |
| Programming Style (Descriptive variable names and proper indentation) | 15 |
| User Interface (Neat prompts and proper spaces and blank lines) | 5 |
| Documentation (Meaningful comments that describe the code) | 10 |
| **Total** | 100 |

\*Two points will be deducted from the total assignment grade for every instruction that is not followed.

**Late work penalty:** 25% if one day late (up to 24 hours late); 50% if two days late (from 24 to 48 hours late); zero credit if more than two days.