Clint Kline

ASD 102 – Object Oriented Programming with C++

10-20-2021

Chapter 5 Programming Exercise 1:

*/\**

*Filename:testFile.cpp*

*Creation Date: 10-13-2021*

*Author: Clint Kline*

*Purpose:*

*- desription.*

*\*/*

#include <iostream>

#include <string>

#include <sstream>

#include <cctype>

using *namespace* std;

string num;

*int* number;

*void* stringToInt(string *str*);

*void* testPos(*int* *num*);

*void* counter(*int* *num*);

*void* seperator(*int* *num*, *int* *count*);

*void* printDigits(*int* *num*);

*void* theSum(*int* *num*);

*int* main()

{

    cout << "\n| | | | | THE SEPERATOR | | | | |" << endl;

    cout << "\n\nPlease, present an offering to the SEPERATOR in the form of an integer : " << endl;

    cout << ">> ";

    cin >> num;

    cout << "\n";

    stringToInt(num);

    main();

    return 0;

};

*void* stringToInt(string *str*)

{

*int* theInt;

    stringstream toInt;

    toInt << *str*;

    toInt >> theInt;

    testPos(theInt);

};

*void* testPos(*int* *num*)

{

    if (*num* > 0)

    {

        counter(*num*);

    }

    else if (*num* < 0)

    {

*num* = *num* \* -1;

        cout << "the SEPERATOR dos not deal in negatives. Your number shall be: " << *num* << "\n"

             << endl;

        counter(*num*);

        theSum(*num*);

    }

    else

        cout << "Do not trifle. The SEPERATOR accepts strictly integers!"

             << "\n\n"

             << "AGAIN!!" << endl;

};

*void* counter(*int* *num*)

{

*int* count = 1;

    cout << "\n"

         << "The SEPERATOR speaks! Behold, your digits : " << endl;

    while (*num*)

    {

        cout << count << ". " << *num* % 10 << endl;

*num* = *num* / 10;

        count++;

    }

    seperator(*num*, count);

};

*void* seperator(*int* *num*, *int* *count*)

{

    cout << "\nThe SEPERATOR has counted " << *count* - 1 << " digits. " << endl;

    cout << "The SEPERATOR does not lie!" << endl;

    printDigits(*num*);

};

*void* printDigits(*int* *num*)

{

*int* newNum;

*int* digit;

    while (*num* > 0)

    {

        digit = *num* % 10;

*num* /= 10;

        cout << digit << endl;

    }

};

*void* theSum(*int* *num*)

{

*int* j;

*int* digit = 0;

*int* sumOfaDig = 0;

    while (*num* > 0)

    {

        sumOfaDig += *num* % 10;

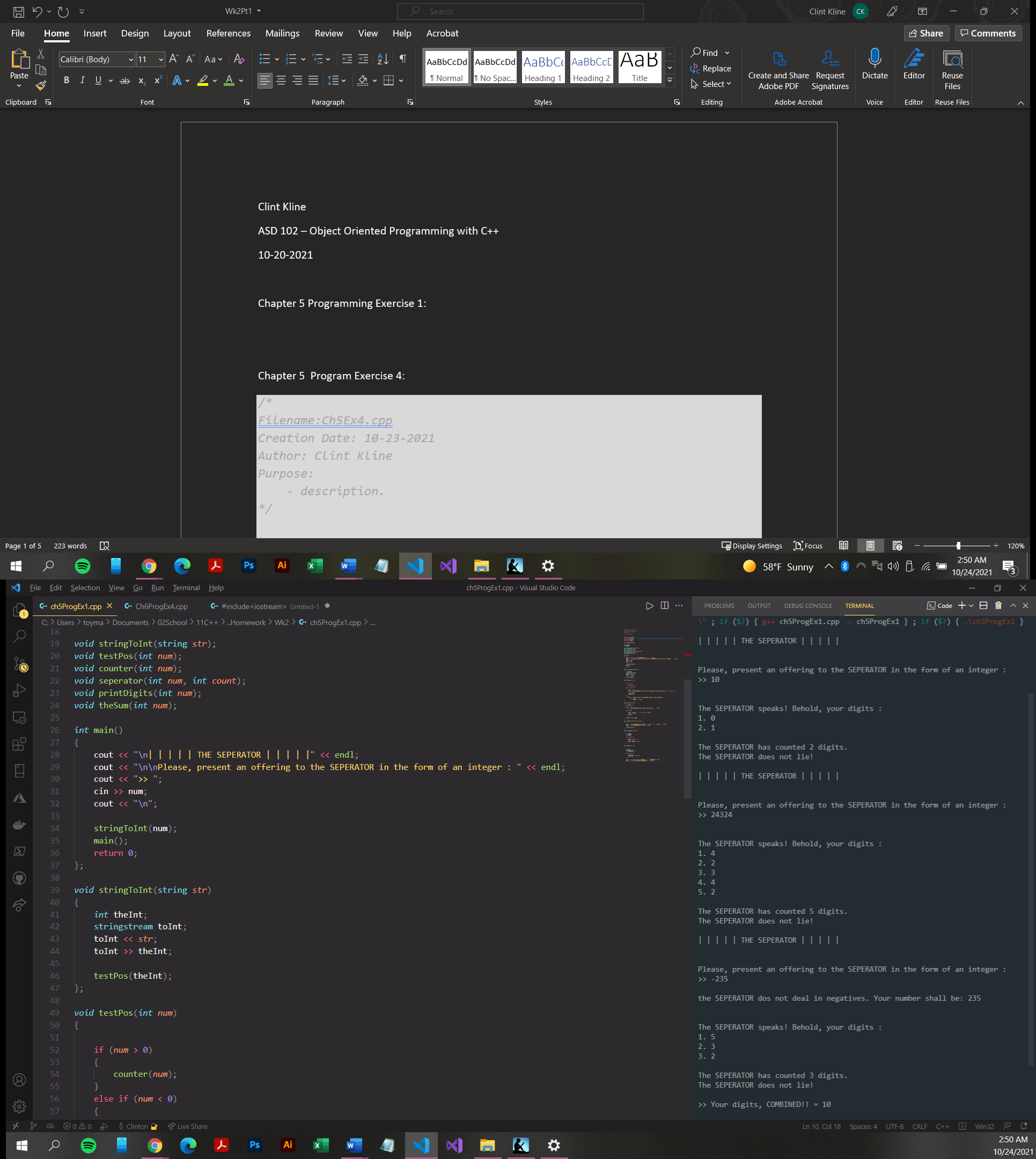
*num* /= 10;

    };

    cout << "\n>> Your digits, COMBINED!! = " << sumOfaDig << endl;

    cout << "\n||| ALL HAIL THE SEPERATOR |||" << endl;

};



Chapter 5 Program Exercise 4:

*/\**

*Filename:Ch5Ex4.cpp*

*Creation Date: 10-23-2021*

*Author: Clint Kline*

*Purpose:*

*- description.*

*\*/*

#include <iostream>

#include <string>

#include <cctype>

#include <vector>

#include <algorithm>

using *namespace* std;

*int* main()

{

    string letterNumber;

    string numberNumber;

*int* count = 0;

    cout << "\n\nEnter a telephone number expressed in letters: " << endl;

    cout << ">> ";

    getline(cin, letterNumber);

*// letterNumber.erase(remove\_if(letterNumber.begin(), letterNumber.end(), isspace()), letterNumber.end());*

    for (*int* i = 0; i <= 7; i++)

    {

*char* digit;

*char* letter = toupper(letterNumber[i]);

        switch (letter)

        {

        case 'A':

        case 'B':

        case 'C':

            digit = '2';

            break;

        case 'D':

        case 'E':

        case 'F':

            digit = '3';

            break;

        case 'G':

        case 'H':

        case 'I':

            digit = '4';

            break;

        case 'J':

        case 'K':

        case 'L':

            digit = '5';

            break;

        case 'M':

        case 'N':

        case 'O':

            digit = '6';

            break;

        case 'P':

        case 'Q':

        case 'R':

        case 'S':

            digit = '7';

            break;

        case 'T':

        case 'U':

        case 'V':

            digit = '8';

            break;

        case 'W':

        case 'X':

        case 'Y':

        case 'Z':

            digit = '9';

            break;

        case ' ':

            letterNumber.erase(remove(letterNumber.begin(), letterNumber.end(), ' '));

            break;

        default:

            cout << "Invalid Entry" << endl;

            break;

        }

        if (count == 3)

        {

            numberNumber += '-';

        }

        else

            numberNumber += digit;

        count++;

    }

    cout << numberNumber << endl;

    main();

    return 0;

};

