**CMSC 140 Programming Project 4**

**Concepts tested in this project**

* Learn to organize code **within a function**
* Learn to pass data to and return data from a function
* **Use of loops**
* Use of output file processing

**Project Description**

Write a program that calculates the average number of days a company's employees are absent during the year and outputs a report on a file named "**employeeAbsences.txt**".

**Project Specifications**

Input for this project:

* the user must enter the number of employees in the company.
* the user must enter as integers for each employee:
  + the employee number (ID)
  + the number of days that employee missed during the past year.

Input Validation:

* **Do not accept a number less than 1** for the number of employees.
* **Do not accept a negative number for the days** any employee missed**.**
* Be sure to **print appropriate error messages** for these items if the input is invalid.

Output: The program should display the following data:

* display a user full name
* display a due date
* display the user full name
* Each employee number (ID) and the number of days missed should be written to the report file named "**employeeAbsences.txt**".
* The average number of days a company's employees are absenting during the year should be written to the report file named "employeeAbsences.txt".

**Processing Requirements**

* Create a **global** variable of type ofstream for the output file. Use this variable to open the file **employeeAbsences.txt** in your program to write data to it. A global variable is defined above the main function and its scope is within the entire program.
* Create the following three functions that will be called by the main function:

1. A function called **NumOfEmployees**. This function asks the user for the number of employees in the company. This value should be returned as an int. The function accepts no arguments (No parameter/input).
2. A second function called **TotDaysAbsent** that accepts an arguments of type int for the number of employees in the company and returns the total of missed days as an int. This function should do the following:
   1. Asks the user to enter the following information for each employee:

* The employee number (ID) (Assume the employee number is 4 digits or fewer, but don't validate it).
* The number of days that employee missed during the past year.
  1. Writes each employee number (ID) and the number of days missed to the output file (**employeeAbsences.txt**). ( Refer to [Sample File Output](#SampleFileOutput) )

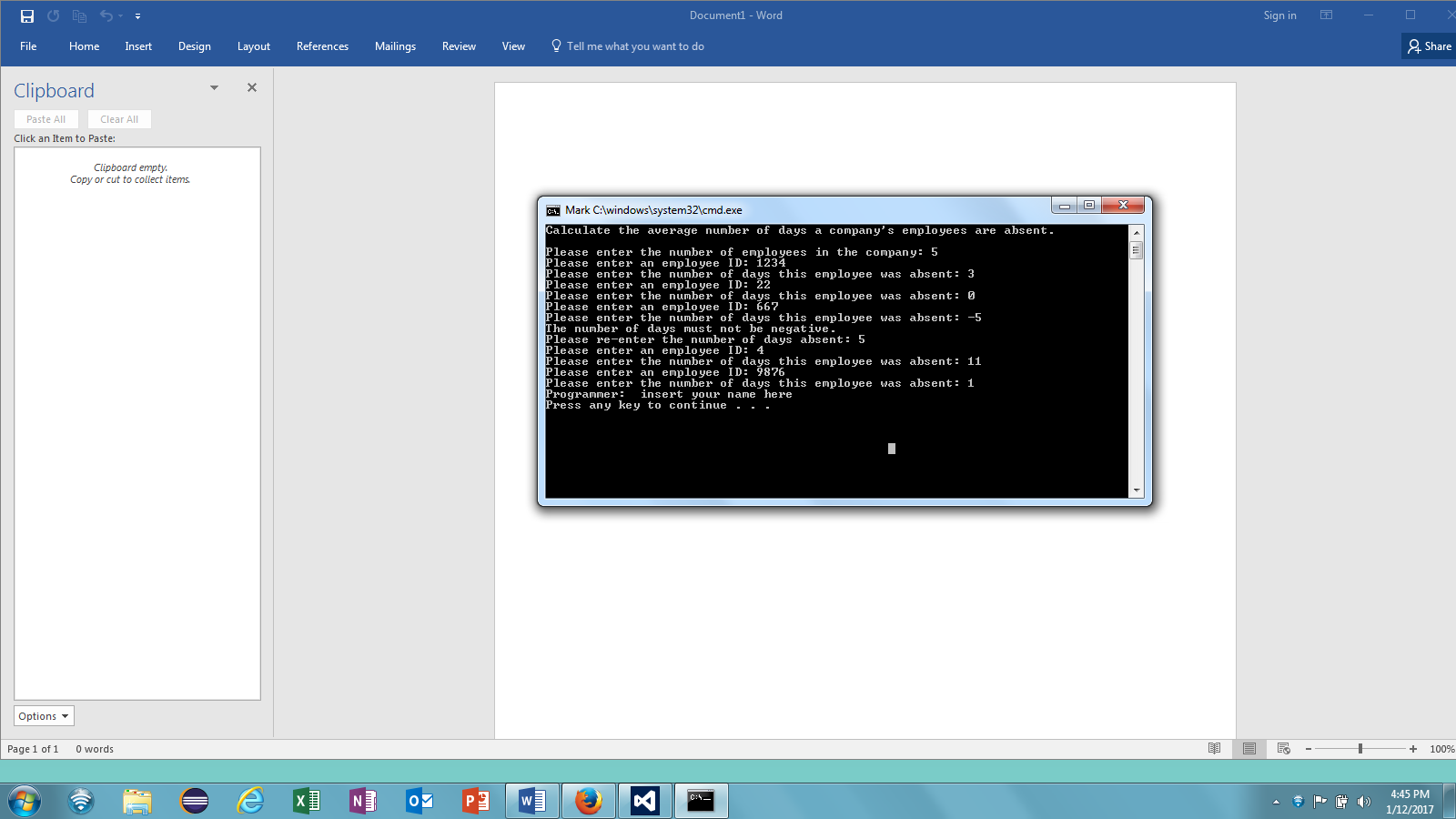
1. A third function called **AverageAbsent** that calculates the average number of days absent.
   1. The function takes two arguments:

* the number of employees in the company
* The total number of days absent for all employees during the year.
  1. This function should return, as a double, the average number of days absent.
  2. This function does not perform screen or file output and does not ask the user for input.

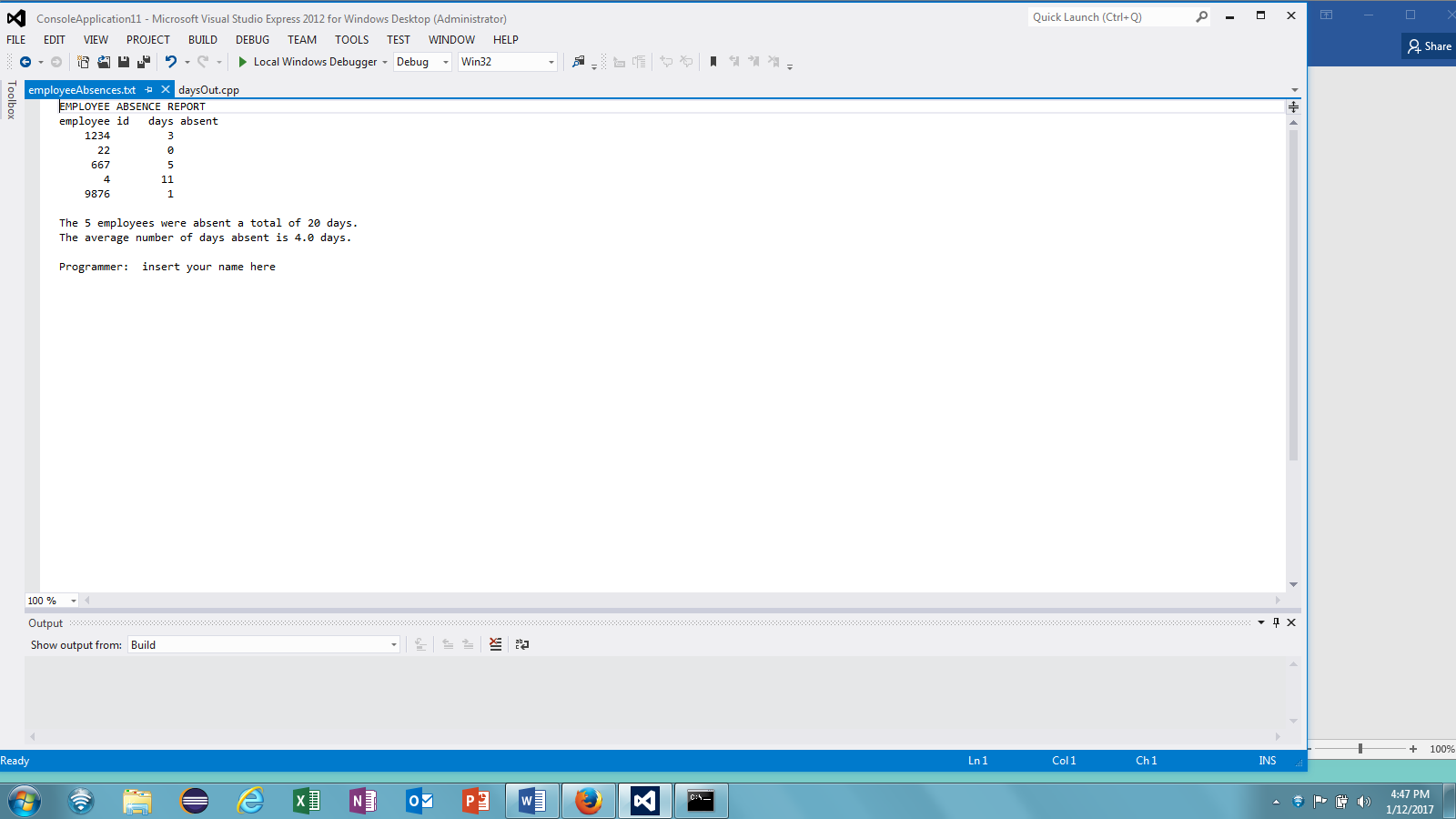
NOTE:

* The location of the file **employeeAbsences.txt** created by your program will be in the same folder as your .cpp source file. Make sure to check your file system directory to see if the file is getting created as the result of running your program and its format matches with the format shown [in the Sample File output format](#SampleFileOutput).
* Also avoid declaring an absolute path for **employeeAbsences.txt,** for example **e:\\ employeeAbsences.txt** or **c:\\cmsc140\\employeeAbsences.txt** in your program**.**

**Sample Screen Output:**



**Sample File Output format:**



**Deliverables:**

Flowchart.

A Test plan : at least with 3 sample cases.

At least with 3 **employeeAbsences.txt** files One for each test case.

At least with 3 output screens. One for each test case.

A word document, pdf or handwritten flowchart.

Last\_NamePr 3.cpp

**NOTE**: Be sure to check also

1. CMSC140 Common Project Submission Requirements (.docx)
2. CMSC140 Grading Rubric\_CheckList-Project4 (.xlsx)

**Test Plan Template**

Test your program with at least 3 more test cases. Use the given data as an example. Record your data for input and output in the following table. **Make sure your tests cover all the possible scenarios.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Input** |  | **Expected Output** | **Actual Output** | **Did the test pass?** |
| 1 | No. of employees :  5  Employee ID’s/Days of absence : 1234 3  22 0  667 **-5**  5  9876 1 |  | “The number of days “ must not be negative  Re-enter ……….  Employee: 5  Total: 20  Average: 4.0 | Output screen  **employeeAbsences.txt** |  |
| 2 | No. of employees :  4  Employee ID’s/Days of absence : 1111 4  1112 5  1113 4  1114 3 |  | Employee: 4  Total: 16  Average: 4.0 | Output screen  **employeeAbsences.txt** |  |
| 3 | No. of employees :  5  Employee ID’s/Days of absence : 1234 2  2234 4  3234 6  4234 9  234 4 |  | Employee: 5  Total: 25  Average: 5.0 | Output screen  **employeeAbsences.txt** |  |
| 4 |  |  |  |  |  |