**Fall 2018 CMSC 140 Programming Project 4: Days Out**

**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 students’ 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 the following three functions that will be called by the main function:

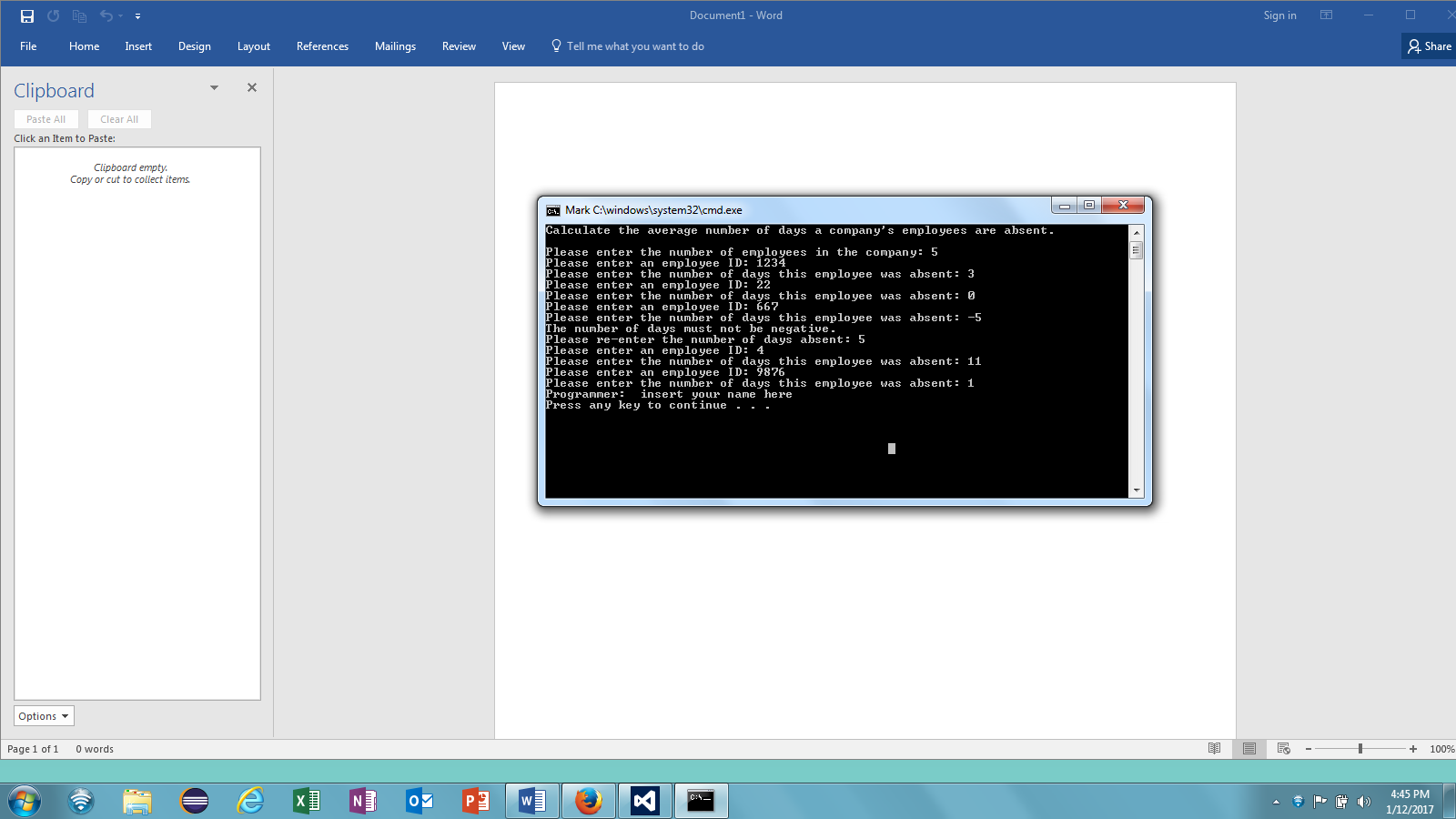
1. A first function asks a user for the number of employees in the company. This value should be returned as an int. The function accepts no arguments.
2. A second function that does the following:
   1. Asks the user to enter for the following information each employee:

* the employee number (ID)
* 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.
  2. The function returns the total of missed days as an int. Assume the employee number (ID) is 4 digits or fewer, but don't validate it.
  3. The function accepts two arguments:
* The name of the file stream object as a reference parameter of type ofstream
* The number of employees in the company as int.

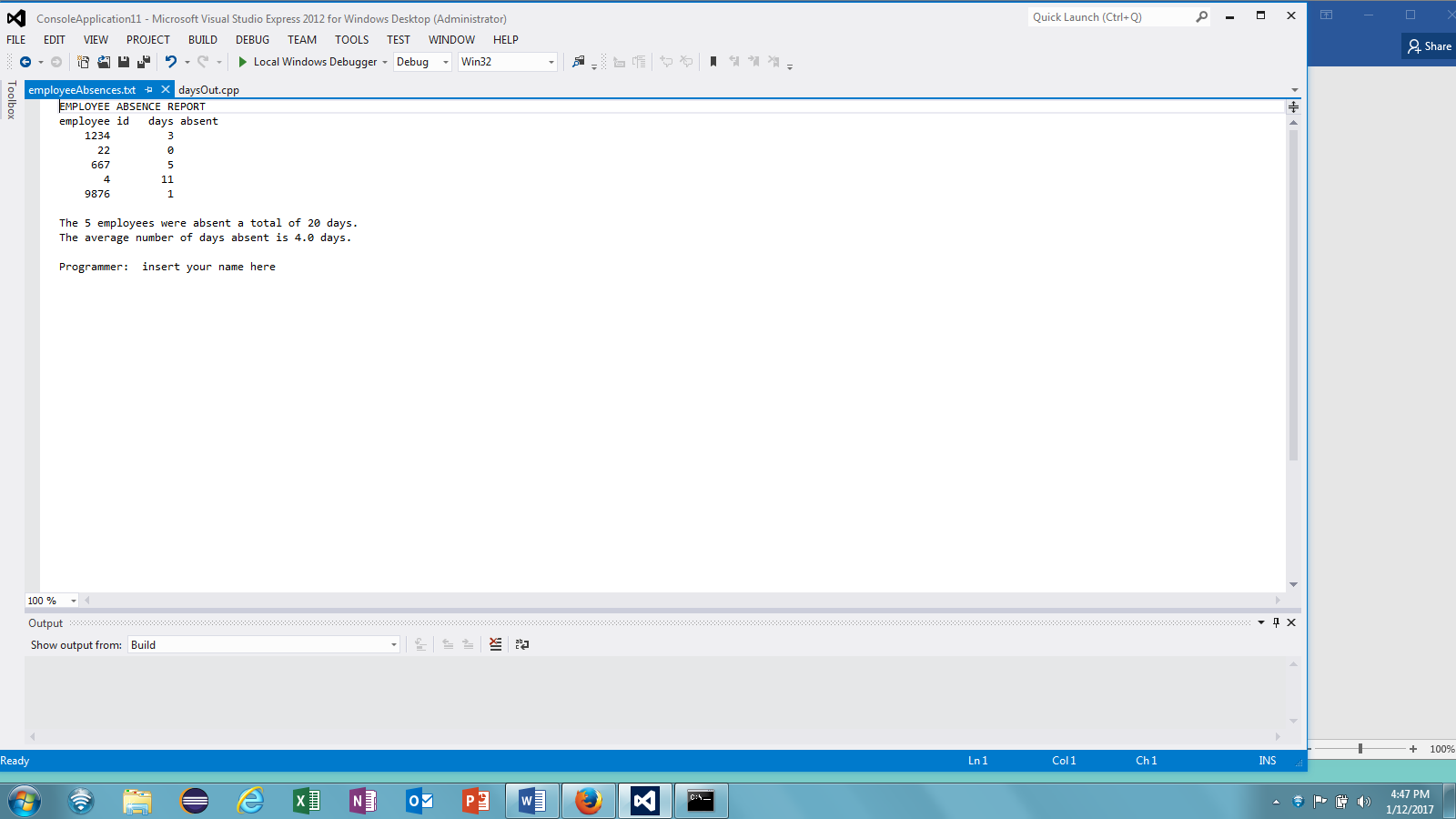
1. A third function calculates the average number of days absent.
   1. The function takes two arguments:

* the number of employees in the company
* the total number of days absent for all employees during the year.
  1. The 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.

**Sample Screen Output:**



**Sample File Output:**



**Project Submission Requirements**

**Deliverables:**

* Flowchart for the Program.
* C++ files (source code)
* Output file
* Documentation file

**Deliverables Format:**

The above deliverables will be packaged as follows:

* C++ files (source code): FirstInitialLastName\_Pr<number>.cpp
* Output file called employeeAbsences.txt
* Documentation: Word document(docx) or .pdf including:
  + Title page including student name, project title and number
  + Flowchart
  + Pseudocode
  + Test plan (table) with at least 2 different data files and test cases for each requirements and screenshots supporting each test case
  + Lessons Learned Section describing any issues you experienced during work on the project and how you solve them.
  + Check List

NOTE:

* All required files should be submitted to the Blackboard Project <number> assignment area no later than the due date.
* If program does not compile, project will get grade “0”. Contact your instructor prior to the project submission due date, if you have compilation issues.

**Project Documentation Requirements**

Program Header: All programming projects’ source code needs to have one block comment at the top of the program containing the course name and CRN, the project number, your name, project description, the due date. Provide any additional comments as necessary to clarify the program.  
Following is a template of the required program header:

/\*

 \* Class: CMSC140 CRN

 \* Instructor:

\* Project4

 \* Description: (Give a brief description for Project)

 \* Due Date:

\* I pledge that I have completed the programming assignment independently.

   I have not copied the code from a student or any source.

   I have not given my code to any student.

   Print your Name here: \_\_\_\_\_\_\_\_\_\_

\*/

Comments: Add comment to variables, formulas, or any part of the program with the purpose of making the source code easier to understand.

Indentation: It must be consistent throughout the program and must reflect the control structure.

Proper naming conventions: Variable and method names need to be descriptive to show the role of the variable or method. Avoid single letter names. Constant names should be all upper-case, variable names should use “camel case” (i.e. start with lower case, with subsequent words starting with upper case: *hours****W****orked* for example) or underscores to separate words (i.e. items\_ordered).

**NOTE**: Be sure to check also

1. CMSC140 Common Project Submission Requirements (.docx)
2. CMSC140 Grading Rubric\_CheckList-Project 4 (.xlsx)

**Test Plan Template**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Input** | **Actual input** | **Expected Output** | **Actual Output** | **Did Test Pass?** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| … |  |  |  |  |  |
| 10 |  |  |  |  |  |