Lab Assignment 1 (Fall 2021)

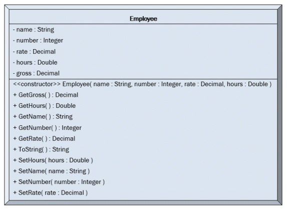
To do this lab, you will need to use **C#** in **Visual Studio Professional 2019**. You can access this program in **Mohawk Apps**, while either on campus or at home. Alternatively, while on campus a local version can be accessed from the **Start Menu**, or, you can download and install it as described by the instructions in the **Student Resources** sub-section located in the **Modules** section of the course page.

To Be Graded – General Details:

- This program will be marked for 6% of your final grade
- Please examine the **Marking Scheme** to see the marks breakdown
- This program needs to have appropriate internal comments, as well as **XML comments** for *every class* and *every method*
- This program also needs to have an appropriate comment block at the top of all code files that contains:
  - Your name and student number
  - The file date
  - The program's purpose
  - Your **Statement of Authorship (https://mycanvas.mohawkcollege.ca/courses/92934/pages/statement-of-authorship)**
- Bundle your project into one Zip file, and upload it to the appropriate **Lab Assignment (https://mycanvas.mohawkcollege.ca/courses/92934/assignments/838001)** on MyCanvas
- Please read about **documentation (https://mycanvas.mohawkcollege.ca/courses/92934/pages/program-documentation)** style
- Programs that are late will be penalized 10% per day (includes each day of a weekend)
- Programs that do not compile or do not include a **Statement of Authorship (https://mycanvas.mohawkcollege.ca/courses/92934/pages/statement-of-authorship)** will be penalized 10% for each

Out of Sorts

Project Name: Lab1   Create Class: Employee



UML class diagram for class Employee

(right-click to view)

Write a console program that:

- Makes use of a class called **Employee** which stores the information for one single employee
  - You **must** use the methods in the UML diagram - You **may not** use class properties

- Reads the data in this csv **employees.txt (https://mycanvas.mohawkcollege.ca/courses/92934/files/17176758/download)** **(https://mycanvas.mohawkcollege.ca/courses/92934/files/17176758/download?download_frd=1)** data file (right-click to save file) into an array of your Employee class
  - There can potentially be any number of records in the data file up to a maximum of 100
  - You **must** use an array of Employees - You **may not** use an ArrayList (or List)
- Prompts the user to pick one of six menu options:
  1. Sort by Employee Name (ascending)
  2. Sort by Employee Number (ascending)
  3. Sort by Employee Pay Rate (descending)
  4. Sort by Employee Hours (descending)
  5. Sort by Employee Gross Pay (descending)
  6. Exit
- Displays a neat, orderly table of all five items of employee information in the appropriate sort order, properly formatted
- Continues to prompt until the user selects the exit option
- The main class (**Lab1**) should have the following features:
  - A **Read()** method that reads all employee information into the array and has exception checking
  - Error checking for user input
  - A **Sort()** method other than a *Bubble Sort* algorithm (You must research, cite and code your own sort algorithm - not just use an existing class method)
  - The **Main()** method should be highly modularized
- The **Employee** class should include proper data and methods as provided by the given UML class diagram to the right
  - No input or output should be done by any part of the **Employee** class itself
  - *Gross Pay* is calculated as *rate of pay* * *hours worked* and after 40 hours overtime is at time and a half
  - Where you calculate the *gross pay* is important, as the data in the *Employee* class should always be accurate
- You may download this **sample program (https://mycanvas.mohawkcollege.ca/courses/92934/files/17176757/download)** for a demonstration of program behaviour

Marking Scheme

Out of Sorts	
Documentation: Comments, Naming Conventions	/ 5
Employee Class: Constructors, Data, Methods	/ 4
Read Method: File I/O, Employee Array, Exceptions	/ 4
Sort Method: Algorithm, Sorts on Fields	/ 3
Menu: All Options, Re-Prompts	/ 2
Output: Neat, Complete	/ 2
Total:	**/ 20**