

# CSCI 1301 - Lab 12

Clément Aubert

February 20, 2018

**Deadlines:** Part I and Part II are required to be able to turn in Project #3, presented in Part III.

## Part I - Comparing Text Files

In this preliminary exercise, we'll study the interest of comparing text files using programs instead of tracking the differences ourselves. One of the goal will be to understand the interest of sharing only the difference between two files instead of sharing the whole file.

### Comparing With Naked Eyes

- a) Download Words1.txt and Words2.txt
- b) Open both files, side by side, and try to find some differences between them. Now, imagine having to do this in a file containing millions of lines.

### Installing and Discovering a Diff Software

- a) There are multiple softwares to automate this task for us, and we will use one of them, called WinMerge.
  - If you are administrator on your computer (that is, if you have the right to install softwares), download this file and execute it.
  - If your are not administrator, then download this “portable version” and execute it.
- b) Once the program is installed, open it by double-clicking on the “WinMergePortable.exe” file stored in the path you indicated at the previous step.
- c) Click on “File”, and then on “Open” (or hit CTRL + o), indicate “Words1.txt” as the left file, and “Words2.txt” as the right file (click on the “Browse” button in both cases to navigate to this file).
- d) You should see something like this:

My Random List	My Random List
Version 1	Version 2
Clément Aubert	Clément Aubert
2/20/2018	2/21/2018
copper	copper
explain	explain
ill-fated	ill-fated
truck	truck
neat	neat
unite	unity
branch	branch
educated	educated
toruous	toruous

Note that every line where there is a difference is highlighted in orange (actually, corn), and that the actual difference is in pale yellow (actually, sidecar).

## Exporting the Difference Between Two Files

- To save only the difference between those two files, click on “Tools”, “Generate Patch...”, and then indicate a file with the .txt extension in the “Results” field, and then click on “Ok”.
- Open the file you generated at the previous step with a text editor, you should see something like

```
2c2
< Version 1
---
> Version 2
4c4
< 2/20/2018
---
> 2/21/2018
11c11
< unite
---
> unity
15c15
< hum
---
> ahem
20c20
< needle
---
> hay
```

If we decompose one of this “block” for instance

```
2c2
```

```
< Version 1
---
> Version 2
```

It is telling us that line 2 of the right file was changed at line 2 of the left file: “Version 1” became “Version 2”.

Note that if you are changing only a couple of characters in a project of millions of lines, sharing only the difference constitutes a huge increase in performance and readability. Indeed, if I have the original code that you modified, I can review directly your modifications without having to track them one by one, and the file that you have to share is significantly lighter!

## Practising

Now, introduce other changes in Words2.txt: delete a line, introduce a blank line, add three words, and change a word. Export the new difference between Words1.txt and Words2.txt: can you read the diff file in all its details? Note that, on top of the c letter that we had previously, other letters are used: can you guess what they stand for? Look for the answer to this question at <https://unix.stackexchange.com/a/216131>.

## Part II - A Class for Chemical Elements

You will study and modify a class for chemical elements (consult [https://en.wikipedia.org/wiki/List\\_of\\_chemical\\_elements#List](https://en.wikipedia.org/wiki/List_of_chemical_elements#List) for a complete list).

## Reading

Download ChemElemProject, extract the project, open it in VS, compile and execute it. Now read the code in “ChemElem.cs” and “Program.cs”.

The class definition “ChemElem.cs” contains:

- Three attributes,
- One constructor,
- One static method,
- One methods that returns the melting point in Celsius,
- One ToString method.

The application program “Program.cs” performs one simple conversion from Kelvin to Celsius, using data given by the user, create and print a ChemElem object (implicitly calling the ‘ToString’ method).

## Modifying

Do the following:

- a) In “ChemElem.cs”, write getters and setters for all attributes
- b) In “Program.cs”, create a second object using our custom constructor, and print the value of its attributes using the getters you previously defined.
- c) In “ChemElem.cs”, write a no-args constructor.
- d) In “Program.cs”, create an object using the no-args constructor, and set its values using the setters you previously defined.
- e) In “Program.cs”, display on the screen the string returned by the ToString method when it is called by the object you created at the previous step.
- f) In “Program.cs”, try calling the FromKelvinToCelsius method with one of your object, for instance using `hydrogen.FromKelvinToCelsius(34)`. What happens?
- g) Still in “Program.cs”, try calling the MeltingInCelsius method with the class, for instance using `ChemElem.MeltingInCelsius()`. What happens?

### Part III - Project #3

Here are the instructions for the Project #3, due the 1st of March, before 11:59PM. Start by downloading a new copy of the ChemElemProject. **Do not start from the version you modified during the previous part, start with a fresh copy of the project.** Perform the following modifications in “ChemElem.cs”:

- Add your name and the date in a delimited comment at the top of the file.
- Add an attribute for the boiling point (in Kelvin).
- Modify the constructor, so that it takes a forth argument, and sets its value to be the value of the boiling point attribute.
- Create a *static* FromKelvinToFahrenheit method, taking inspiration from the FromKelvinToCelsius method.
- Create a MeltingInFahrenheit method, that returns the melting point in Fahrenheit of the calling object. This method should use your FromKelvinToFahrenheit method.
- Create a BoilingInFahrenheit method, that return the boiling point in Fahrenheit of the calling object. This method should use your FromKelvinToFahrenheit method.
- Modify the ToString method, so that the string returned includes
  - The name of the chemical element and its atomic number,
  - The melting point in Kelvin and in Fahrenheit,
  - The boiling point in Kelvin and in Fahrenheit.

You should test all of those modifications in your “Program.cs” file as you implement them. Use relevant data, test your program, and make sure the behaviour is the expected behaviour.

Once you are done,

- a) Re-download another copy of ChemElemProject, and extract it.
- b) Open WinMerge, the program you used in Part I.
- c) Load as the left file the original “ChemElem.cs”, and as the right file the “ChemElem.cs” you obtained by doing all the modifications listed above.
- d) Export the difference between those two files as a .txt file, named “Lname\_Fname.txt”, where

Lname (resp. Fname) is your last name (resp. first name).

- e) Send me the “Lname\_Fname.txt” file **and only this file** (no .cs file, no project, no zip archive) *via* email before the deadline.