Department of Computer Science and Information Technology
La Trobe University

CSE1OOF/4OOF Summer 1, 2018
Assignment Part B

8%

This is an individual Assignment. You are not permitted to work as a Pair Programming partnership or any other group when writing this assignment.

Due Date

Due : Thursday 13th of December 2018 at 10.00 a.m.
Execution test: Friday 14th of December 2018 at 1.00 p.m.

Delays caused by computer downtime cannot be accepted as a valid reason for a late submission without penalty. Students must plan their work to allow for both scheduled and unscheduled downtime. **There are no days late or extensions on this assignment as execution test marking will begin on** Friday 14th of December 2018 **– in your normal lab (Week 6).** After the submit server has closed, NO assignments can be accepted. Non-attendance at the week 4 lab you have signed up for on the LMS page will also result in your assignment being awarded 0, except as detailed below.

Copying, Plagiarism

Plagiarism is the submission of somebody else's work in a manner that gives the impression that the work is your own. The Department of Computer Science and Information Technology treats academic misconduct seriously. When it is detected, penalties are strictly imposed. Refer to the subject guide for further information and strategies you can use to avoid a charge of academic misconduct.

Assessment Objectives

- To practise using the **String** class. (Lecture/Workshop 2)
- To practice using switch and/or multiway selection branches
- To practise reading from and writing to the console and reading from a text file. (Lecture 4, Lab 3)
- To practise using loops (repetition)

Submission Details and marking

Full instructions on how to submit electronic copies of your source code files from your latcs8 account are given on **page 2**. All assignments in OOF are marked, face to face, in the lab, in an execution test. This means that we mark running code. Your code must compile and display a result to the screen. Regrettably, we don't have the time or resources to look at code. The smallest amount of code that produces and displays a correct result will gain more marks than lots of code that doesn't compile, run or display something to the screen. If you cannot attend the lab you have signed up for on the LMS page, please email me (m.felicetti@latrobe.edu.au) to arrange another time.

Marking summary

This assignment is worth **8%** of your final mark in this subject.
Implementation **(Execution of code) 80%, explanation of code 20%**

Instant zeros or heavily reduced marks	**Not submitting code** **Using arrays** **Not attending marking session** **Not able to explain code that has not been taught yet** **Does not run on latcs8** **Uses System.exit()**
ChemicalCompounds**.java**	**100% (80% Execution and 20% Explanation)**

Using code not taught in OOF

Please also note carefully that whilst we encourage innovation and exploring java beyond what has been presented in the subject to date, **above all, we encourage understanding**.

All of the Tasks that follow can be solved using techniques that have been presented in lectures, lecture / workshops and labs so far.

These are the techniques and knowledge that we will later be examining in the Real Time Test (20 marks) and the exam (60 marks).

Code and techniques that are outside the material presented will not be examined, of course.

For this task you cannot use arrays

This assignment is designed to demonstrate the techniques that have been presented in lectures, lecture / workshops and labs so far. You may only use the following classes:

Math, System, String, Scanner and File

Submitting an assignment with code outside what has been presented so far and not attending the marking execution test will result in an automatic mark of 0, regardless of the correctness of the submission.

Electronic Submission of the Source Code

- Submit all the Java files that you have developed in the tasks above.
- The code has to run under Unix on the latcs8 machine.
- You submit your files from your latcs8 account. Make sure you are in the same directory as the files you are submitting. Submit each file separately using the **`submit`** command.

submit OOF ChemicalCompounds.java

After submitting the files, you can run the following command that lists the files submitted from your account:

verify

You can submit the same filename as many times as you like before the assignment deadline; the previously submitted copy will be replaced by the latest one.

Please make sure that you have read page 1 about the submission close off date and time and the compulsory requirement to attend the execution test in Week 4

Failure to do both of these things will result in your assignment be awarded a mark of 0, regardless of the correctness of the program.

Execution test marks are provisional and subject to final plagiarism checks and checks on the compliance of your code to this assignment document.

As such, final assignment marks may be lower or withdrawn completely.

Background - Task

You are tasked with writing **one** Java program related to chemical compounds and elements. The task will use a file which will contain information on all known chemical elements. In the task, given a set of compounds you will break the chemical formula up to express the number and types of chemical elements, whilst also detecting particular errors in the formula.

Background – Chemical Elements

A chemical element is a type of atom that make up all the ordinary mater in the universe, and is define as an atom that has the same number of protons in its nucleus. Currently there are 118 known elements. These 118 can be represented on the periodic table of chemical elements as shown below:

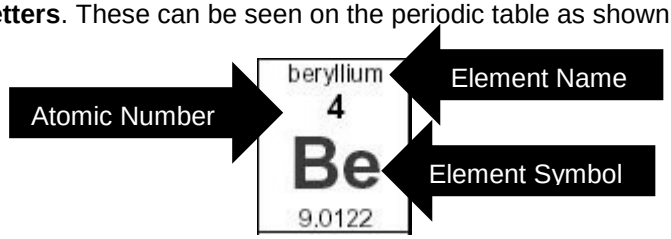
hydrogen 1 H 1.0079																	helium 2 He 4.0026
lithium 3 Li 6.941	beryllium 4 Be 9.0122											boron 5 B 10.811	carbon 6 C 12.011	nitrogen 7 N 14.007	oxygen 8 O 15.999	fluorine 9 F 18.998	neon 10 Ne 20.180
sodium 11 Na 22.990	magnesium 12 Mg 24.305											aluminium 13 Al 26.982	silicon 14 Si 28.086	phosphorus 15 P 30.974	sulfur 16 S 32.065	chlorine 17 Cl 35.453	argon 18 Ar 39.948
potassium 19 K 39.098	calcium 20 Ca 40.078	scandium 21 Sc 44.956	titanium 22 Ti 47.867	vanadium 23 V 50.942	chromium 24 Cr 51.996	manganese 25 Mn 54.938	iron 26 Fe 55.845	cobalt 27 Co 58.933	nickel 28 Ni 58.693	copper 29 Cu 63.546	zinc 30 Zn 65.39	gallium 31 Ga 69.723	germanium 32 Ge 72.61	arsenic 33 As 74.922	selenium 34 Se 78.96	bromine 35 Br 79.904	krypton 36 Kr 83.80
rubidium 37 Rb 85.468	strontium 38 Sr 87.62	yttrium 39 Y 88.906	zirconium 40 Zr 91.224	niobium 41 Nb 92.906	molybdenum 42 Mo 95.94	technetium 43 Tc [98]	ruthenium 44 Ru 101.07	rhodium 45 Rh 102.91	palladium 46 Pd 106.42	silver 47 Ag 107.87	cadmium 48 Cd 112.41	indium 49 In 114.82	tin 50 Sn 118.71	antimony 51 Sb 121.76	tellurium 52 Te 127.60	iodine 53 I 126.90	xenon 54 Xe 131.29
caesium 55 Cs 132.91	barium 56 Ba 137.33	57-70 *	lutetium 71 Lu 174.97	hafnium 72 Hf 178.49	tantalum 73 Ta 180.95	tungsten 74 W 183.84	rhenium 75 Re 186.21	osmium 76 Os 190.23	iridium 77 Ir 192.22	platinum 78 Pt 195.08	gold 79 Au 196.97	mercury 80 Hg 200.59	thallium 81 Tl 204.38	lead 82 Pb 207.2	bismuth 83 Bi 208.98	polonium 84 Po [209]	astatine 85 At [210]	radon 86 Rn [222]
francium 87 Fr [223]	radium 88 Ra [226]	89-102 * *	lawrencium 103 Lr [262]	rutherfordium 104 Rf [261]	dubnium 105 Db [262]	seaborgium 106 Sg [266]	bohrium 107 Bh [264]	hassium 108 Hs [269]	meitnerium 109 Mt [268]	ununnilium 110 Uun [271]	ununnilium 111 Uuu [272]	ununbium 112 Uub [277]		ununquadium 114 Uuq [289]			

*Lanthanide series	lanthanum 57 La 138.91	cerium 58 Ce 140.12	praseodymium 59 Pr 140.91	neodymium 60 Nd 144.24	promethium 61 Pm [145]	samarium 62 Sm 150.36	europium 63 Eu 151.96	gadolinium 64 Gd 157.25	terbium 65 Tb 158.93	dysprosium 66 Dy 162.50	holmium 67 Ho 164.93	erbium 68 Er 167.26	thulium 69 Tm 168.93	ytterbium 70 Yb 173.04
**Actinide series	actinium 89 Ac [227]	thorium 90 Th 232.04	protactinium 91 Pa 231.04	uranium 92 U 238.03	neptunium 93 Np [237]	plutonium 94 Pu [244]	americium 95 Am [243]	curium 96 Cm [247]	berkelium 97 Bk [247]	californium 98 Cf [251]	einsteinium 99 Es [252]	fermium 100 Fm [257]	mendelevium 101 Md [258]	nobelium 102 No [259]

Periodic Table
Source: LeVanHan
Creative Commons

Background – Atomic Numbers and Element Symbols

Each element has an atomic number, these numbers are equal to the number of protons in each atom and hence defines the element. **The elements are all given names but are often represented as a series of letters (element symbol), starting with an Upper-case letter and following with zero, one or two lower case letters**. These can be seen on the periodic table as shown below.



Beryllium
Source: LeVanHan
Creative Commons

Background – “element_list.csv”

A list of all known elements have been put into a file for you named “element_list.csv” and is guaranteed to be without errors and 118 lines long. **(Note: The elements are not in Alphabetical order)**
Each line contains one element and is of the format:

```
A,B,C
```

Where A is the element symbol, B is the element name and C is the atomic number.

An extract is shown below

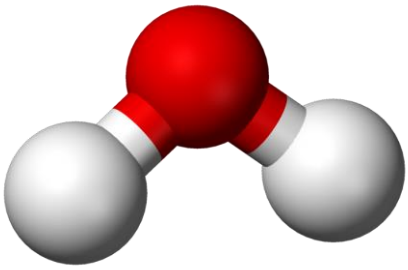
```
Ac,Actinium,89
Al,Aluminum,13
Am,Americium,95
Sb,Antimony,51
Ar,Argon,18
As,Arsenic,33
At,Astatine,85
Ba,Barium,56
Bk,Berkelium,97
Be,Beryllium,4
Bi,Bismuth,83
Bh,Bohrium,107
B,Boron,5
Br,Bromine,35
Cd,Cadmium,48
Ca,Calcium,20
```

This file can be copied from the csilib area

```
cp /home/student/csilib/cse1oof/assignSum18/assignB/element_list.txt .
```

Background – Chemical compounds

A chemical compound or molecule is collection of atoms of more than one chemical element held together by chemical bonds. A common example of this is pure water, which is made up of two Hydrogen atoms and one Oxygen atom that are chemically bound.



Model of a Pure Water Molecule
Public Domain

Background – Chemical formula (molecular formula)

A chemical formula lists the elements and the proportion of atoms per element using element symbols. More specifically the chemical formula written as a molecular formula includes the exact number of atoms per element using element symbols. This is done by listing the symbols and a subscript to indicate the number of atoms for a particular atom (with no subscript indicating one).

Example:

$$H_2O$$

Looking at our periodic table we can see the symbol H stands for Hydrogen, and O for oxygen. The subscript 2, indicates that there are two Hydrogen atoms and one Oxygen atom.

Background – Hill notation

A molecular formula can be written in what is known as Hill notation. This notation specifies that that if the molecule contains Carbon (C) atoms then it must be indicated first, then the number of Hydrogen (H) atoms next if any, and then the rest of the chemical elements in alphabetical order. If there is no Carbon atom then all the chemical elements must be in alphabetical order. Furthermore, single letters come before two or three letter symbols beginning with the same letter (eg. B comes before Be)

Examples:

C_3BBe_4F	is in Hill notation
BBe_4CaF	is in Hill notation (note there is no Carbon, Ca is Calcium)
CH_2BBe_4Ca	is in Hill notation (note as Carbon is included Hydrogen follows next)
BBe_4CaH_2	is in Hill notation (note as Carbon is not included Hydrogen is after Ca)
H_2O	is in Hill notation

BBe_4C_3F	is **NOT** in Hill notation (Carbon does not come first)
C_3FBBe_4	is **NOT** in Hill notation (F should come after Be)
C_3Be_4BF	is **NOT** in Hill notation (B should come after Be)
$FBBe_4Ca$	is **NOT** in Hill notation (F should come after Ca)
$CBBe_4CaH_2$	is **NOT** in Hill notation (H should come after C)
H_2BBe_4Ca	is **NOT** in Hill notation (H should come after Ca, as there is no carbon)
OH_2	is **NOT** in Hill notation

Background – Input file

The input file will contain a list of Chemical compounds (Molecules).
Each line will contain one compound (guaranteed) and will be in the format:

A,B

Where A is the compound written as a molecular formula in Hill notation and B is the name of the compound.
As we cannot use subscript, the number of atoms, will be always to the right of the element symbol.

Example:

H_2O will be written as H2O
BBe_4CaH_2 will be written as BBe4CaH2

Errors:

It is expected that not all molecular formulas will be correct, the errors include:

- Element symbols that do not exist eg. Hr
- Not in Hill notation

Background - Example input files

Two example input file for Task 1 may be copied from the csilib area

```
cp /home/student/csilib/cse1oof/assignSum18/assignB/chemicals1.txt .
cp /home/student/csilib/cse1oof/assignSum18/assignB/chemicals2.txt .
```

Be aware these do not cover all cases, hence you are likely to want to make your own test input files

Background - Limitations and Notes

Error checking – Element list file
In this program you can assume that the data in the element list is without errors and are in the specified format.
Number of records – Element list file
The element list you can guarantee will have 118 lines (as that is the number of known elements)
File name – Element list file
The element list file will always be “element_list.txt”, i.e. **you can** hard code the file name.

Error checking – Input file
The data in the input file will contain errors but limited to those specified in the task.
Number of records – Input file
The input file may have 1 to any number of records.
File name – Input file
The input file must accept any file name, i.e. **do not** hard code the file name.

Task 1 - Solution length

The code should be around 150-250 lines of code. If your code is significantly larger, you may want to reconsider your approach.

Task 1 – ChemicalCompounds.java

Follow the steps to complete the task using the steps and information above. (Note indentation indicates that the indented parts are contained or related to the previous step. Eg. All steps within an if statement are indented)

The task is to read chemical compounds from a file. Then, for each chemical determine if each of the elements are valid and determine if the chemical compound is in correct Hill notation. If valid then information on the compound is printed else the reason it is invalid is printed.

Task 1 – Stage 1

We are going to set up the main components of the program to start, validate it works, then go back and add more code.

1. The program prints to the screen the student number, student name and the assignment number. This is enclosed by double asterisks. Ensure you follow the below format.

   ```
   *** John Smith 11117777 ***
   ```
   (Replace with your details)
2. The program prints the message shown below as to indicate the programs purpose

   ```
   Chemical Compound Information
   ```
3. Prompt the user to enter the input file name
4. Open the input file and set up an appropriate Scanner object for reading.
5. Iterate through each line (chemical compound) of the input file **while** the file has a next line (**See "How to read in from a text file" on LMS**)
   a. Open the "element_list.csv" file and set up an appropriate Scanner object for reading
   b. Store the name of the compound and the compound formula in two separate variables, by using the **substring** and **indexof** methods.
   c. Set up variables:

      An appropriate variable for counting the number of elements initialized to 1
      A String variable for storing the current element
      A String variable for storing the previous element initialized as an empty string
      A String variable for storing the current element name initialized as an empty string
      A String variable for storing the current elements atomic number initialized as an empty string
      A String variable for storing the number of atoms of current element initialized as an empty string
      An appropriate variable to indicate that the compound is valid, initialized to true
      An appropriate variable to indicate Carbon is the first element, initialized to false
      A String variable for storing the invalid reason initialized as an empty string
      A String variable for storing the output valid result (outMessage) initialized as an empty string
   d. Store the first char of the line into the variable for your current element
   e. Iterate by 1 from 1 using variable named "i", to the compound length +1 (exclusive) whilst the compound is valid
      (We start at one as we have already stored the first (0 index) part of or full element)
      i. After completing Stage 1, in this loop complete **Hill notation and validation** steps as set out below.
   f. After iterating through the line
      i. If the compound is valid then print the compound name, the compound formula in round brackets and the outMessage as shown in the **output example**
      ii. Else if the compound is invalid then print "Invalid chemical formula: " and the invalid reason as shown in the **output example**
6. After iterating through the file, close both files and end the program appropriately.

Task 1 – Stage 2 - Hill notation and validation

1. If the next character (index i character) in the input line is a lower case letter, (And i is not equal to the length of reactants String) then concatenate it to the current element. Eg. if input is “Zn”, then before iterating “Z” would be stored in the current element, then character at index 1 is lower case, hence current element would now store “Zn”. (You also need to consider the case when there are no more characters)
2. Else if the next character is a number, (And i is not equal to the length of reactants String) then concatenate it to the number of atoms of current element. Eg. If the input is “Zn34”, then initially “Z” would be stored as the current element and empty as the number of atoms, then on the next iteration “Zn” would be stored as the current element. Then on the next iteration “3” would be stored in the number of atoms, and following this “34” in the number of atoms. (You also need to consider the case when there are no more characters)
3. Else (You have the element symbol and the number of atoms (if blank, number of atoms is 1)
    a. Iterate through the “element_list.csv” and try to find the matching element using the background information (No more information is given here).
        i. If found store the current element name and the current elements atomic number
        ii. Else if not found set the variable to indicate that the compound is valid to false AND store the invalid reason as “No such element.”
    b. If the compound is valid check if the compound is in Hill notation using the background information (Hint. Use the previous element and current element) (No more information is given here)
        i. If it is not in Hill notation set the variable to indicate that the compound is valid to false AND store the invalid reason as the appropriate reason:

            “Not in Hill notation – Carbon not listed first.”
            “Not in Hill notation – Hydrogen not listed after Carbon.”
            “Not in Hill notation – Alphabetical order incorrect”
    c. If the compound is valid then
        i. If the it is the only element (you can work this out using your iteration variable) then concatenate to the outMessage the number of elements, a space, the elementName, the element symbol in round brackets, the atomic number in round brackets and
            1. If more than one atom then “atoms.”
            2. Else “atom.”
        ii. If it is the last element of more than one element, do the same as above but add “and “ before the element name.
        iii. Else:
            1. Concatenate to the outMessage the number of elements, a space, the elementName, the element symbol in round brackets, the atomic number in round brackets and
            2. If more than one atom then “atoms,” (note comma)
            3. Else “atom,” (note comma)
            4. Then store the current element in the previous element variable.
            5. Set the Strings current element, number of elements, number of atoms and atomic number each to empty stings
            6. Store the next character (index i character) in the current element
            7. Increment the element count

Task 1 - Example input file 1

Given the file named “chemicals1.csv”

```
B2Be,Beryllium boride
Ac2O3,Actinium(III) oxide
C3HN,Cyanopolyyne
C3H3N,Azete
N2O,Nitrous oxide
```

Task 1 - Example output 1

The output below is produced, given the inputs (green)

```
*** 37777777 John Smith ***
Chemical Compound Information
Enter the chemical compounds file name: chemicals1.txt

Beryllium boride (B2Be) ::: 2 Boron(B)(5) atoms, and 1 Beryllium(Be)(4) atom.
Actinium(III) oxide (Ac2O3) ::: 2 Actinium(Ac)(89) atoms, and 3 Oxygen(O)(8) atoms.
Cyanopolyyne (C3HN) ::: 3 Carbon(C)(6) atoms, 1 Hydrogen(H)(1) atom, and 1 Nitrogen(N)(7) atom.
Azete (C3H3N) ::: 3 Carbon(C)(6) atoms, 3 Hydrogen(H)(1) atoms, and 1 Nitrogen(N)(7) atom.
Nitrous oxide (N2O) ::: 2 Nitrogen(N)(7) atoms, and 1 Oxygen(O)(8) atom.
```

Task 1 - Example input file 2

Given the file named “chemicals2.csv”

```
Kx3O4P,Tripotassium phosphate
Sf2Mo,Molybdenum sulfide
OC,Carbon monoxide
BeB2,Beryllium boride
O3Ac2,Actinium(III) oxide
C3NH,Cyanopolyyne
H3C3N,Azete
```

Task 1 - Example output 2

The output below is produced, given the inputs (green)

```
*** 37777777 John Smith ***
Chemical Compound Information
Enter the chemical compounds file name: chemicals2.txt

Invalid chemical formula: No such element.
Invalid chemical formula: No such element.
Invalid chemical formula: Not in Hill notation - Carbon not listed first
Invalid chemical formula: Not in Hill notation - Alphabetical order incorrect
Invalid chemical formula: Not in Hill notation - Alphabetical order incorrect
Invalid chemical formula: Not in Hill notation - Hydrogen not listed after Carbon
Invalid chemical formula: Not in Hill notation - Carbon not listed first
```

Transferring files between Windows and Unix

Be very careful transferring files from Windows to Unix. If you do transfer a file from Windows to Unix open the file, in Unix, using vi.
For example, if you transferred a file named **`b.txt`** from Windows to Unix

open the file in Unix with the command

vi –b b.txt

you will (might) see a lot of **`^M`**'s at the end of each line.

These MUST be removed using the command shown below or else your input file will have too many newline characters and will not translate properly. That is, your code will not correctly read the input file for Tasks 1 and 2.

Your code will work on Windows but NOT on Unix.

Still in vi, in command mode (press the Esc key first) do the following

:%s/ctrl-v ctrl-m//g

`ctrl-v ctrl-m` means hold down the control key and with the control key down press v then press m.

Final notes

There will be consultation sessions for the assignment, the times will be posted on LMS, if you have problems come to consultation.

Do a little bit every night; before you know it you will be finished. The assignment is marked with running code, so you are better to have 1 or 2 Tasks completed that actually compile and run, rather than a whole lot of code that doesn't compile.

The execution test is done on latcs8 so please make sure that your code runs on latcs8 before you submit.