**Assignment 3**

**Due date**

* 11.59 PM on ~~March~~ April ~~30th~~ ~~31st~~ 2nd.

Submit your code as per the provided instructions. A signup sheet will be provided to you during class to setup an appointment with the TA to provide a demo of your project.

**Updates**

* Wed Mar 22 22:31:17 EDT 2017: directory structure

**Assignment Goal**

Application of design principles to implement a spreadsheet feature

**Team Work**

* CS 542: team work is NOT allowed. Work individually.
* CS 442: teams of two students.

**Programming Language**

You are required to program using Java.

**Compilation Method**

* You are required to use ANT for compilation of code written in Java.
* Your code should compile and run on *remote.cs.binghamton.edu*or the *debian-pods*in the Computer Science lab in the Engineering Building.

**Policy on sharing of code**

 EVERY line of code that you submit in this assignment should be written by you or be part of the code template provided for this assignment. Do NOT show your code to any other group. Our code-comparison software can very easily detect similarities.

 Post to the listserv if you have any questions about the requirements. Do NOT post your code to the listserv asking for help with debugging.

**Project Description**

* From the command line, accept the following as input, in this order:
  + The name of the input file (referred to as input.txt below)
  + The name of the output file (referred to as output.txt below)
  + Debug Value: an integer that controls what is printed on stdout
    - Validate that the correct number of command line argumets have been passed.
    - Validate that the DEBUG\_VALUE is between 0 and 4.
  + The file ~~Input.java~~ input.txt will be provided to you that has a series of assignments. ~~to a 2D array, along with statements to print the value of a certain cell. The 2D array is used a simple representation of a spreadsheet. [This class will be designed in class on 3/16/17]~~.
  + Here is an example of the input.txt file
  + a1=32+27
  + a2=a1+37
  + a3=a1+a2
  + b1=29+31
  + a1=37+b1
  + Each line of the input file will have a simple expression. The value before the equals sign will always refer to the cell. The right hand side will always have two operands and the "+" operator. The operands will be either two digit integers greater than 10 or cell indexes. The cell indexes will only refer to rows a-z, and columns 1-26.
  + You need to use the Observer Pattern to implement the feature of a spreadsheet wherein values of cells get updated, when the cells they are dependent on get updated.
  + Use an appropriate data structure in the SubjectImpl to store the cells. Justify your choice of data strucure in README.txt.
  + Before a cell is updated, detect if there is a cycle. If so, print a meaningful error message in output.txt explaining how the cycle is formed, and discard that update. Process the next update. In your README.txt justify your choice of the cycle detection algorihm.
  + The last line of output.txt should be: The sum of all cell values is: X
  + Do NOT use Java's in-built Observer Pattern classes/interfaces. Design your own.
  + The choice of data structures and algorithms used should be justified in the README.txt in terms of space and/or time complexity.
  + Use Singleton for your Logger.
  + The DEBUG\_VALUE, read from the command line, should be set in the Logger class. Use the DEBUG\_VALUE in the following way:
    - DEBUG\_VALUE=4 [Choose your own criteria]
    - DEBUG\_VALUE=3 [Choose your own criteria]
    - DEBUG\_VALUE=2 [Choose your own criteria]
    - DEBUG\_VALUE=1 [Choose your own criteria]
    - DEBUG\_VALUE=0 [No output should be printed from the application to stdout]

**Sample Input Files sent by students in this course**

Please note that I have not verfied these input files.

* From Alex. Thanks!
  + [input-1](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/assignments/assign3/input-1.txt) [The sum of all cell values is: 884]
* From Yu H Chen. Thanks!
  + [fibSpreadsheet](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/assignments/assign3/fibSpreadsheet.txt) [The sum of all cell values is: 8320370]
  + [lucasSpreadsheet](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/assignments/assign3/lucasSpreadsheet.txt) [The sum of all cell values is: 18604910]
  + [longCyclesSpreadsheet.txt](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/assignments/assign3/longCyclesSpreadsheet.txt) [The sum of all cell values is: 220545]. Example output file [The explanation beyond Cycle detected will be appreciated, but not required] [output-long-cycles.txt](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/assignments/assign3/output-long-cycles.txt)

**Code Organization**

* Use the directory structure below, and add new packages/classes/interfaces, as needed.
* firstName\_lastName\_assign\_2/spreadsheetUpdates
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src/BUILD
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src/BUILD/classes
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src/spreadsheetUpdates
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src/spreadsheetUpdates/driver
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src/spreadsheetUpdates/driver/Driver.java
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src/spreadsheetUpdates/observer
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src/spreadsheetUpdates/observer/Subject.java
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src/spreadsheetUpdates/observer/Listener.java
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src/spreadsheetUpdates/util
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src/spreadsheetUpdates/util/FileProcessor.java
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src/spreadsheetUpdates/util/Logger.java
* firstName\_lastName\_assign\_3/spreadsheetUpdates/src/build.xml
* firstName\_lastName\_assign\_3/README.txt
* Use the following commands to un-tarzip the code template (from assignment-2).
* tar -zxvf john\_doe.tar.gz
* If you are working on a team project, change the top level directory to firstName\_lastName\_firstName\_lastName\_assign3

**Submission**

* Make sure to first run "ant clean" to remove the BUILD folder
* Delete the input and output files from your folder
* At the level above john\_doe\_assign\_3 run the following commands
* tar -cvf john\_doe\_assign\_3.tar john\_doe\_assign\_3/
* gzip john\_doe\_assign\_3.tar
* Alternatively, at the level above john\_doe\_assign\_3 run the following commands
* tar -czvf john\_doe\_assign\_3.tar.gz john\_doe\_assign\_3
* Read [this](http://www.cs.binghamton.edu/~mgovinda/courses/README.txt) file for general guidelines on how to prepare a README for your submission.
* Run "ant clean" and make sure all class files, object files (.o files), executables, and backup files are deleted before creating a zip or tarball. To create a tarball, use the provided (or your own) target in build.xml. The command to "untar" should be specified in the README.txt.
* Both the team members should submit.

**General Requirements**

* Start early and avoid panic during the last couple of days.
* Submit a README.txt file (placed at the top level). The README.txt file should be filled out as described in that file.
* Apply all design principles (wherever applicable).
* Separate out code appropriately into methods, one for each purpose.
* You should document your code. The comments should not exceed 72 coloums in width. Use javadoc style comments if you are coding in Java.
* Do not use "import XYZ.\*" in your code. Instead, import each required type individually.
* Every class that has data members, should have corresponding accessors and mutators (unless the data member(s) is/are for use just within the method.).
* If a class does not implement an interface, you should have a good justification for it. For example, it is ok to have an abstract base class and a derived class, wherein both do not implement interfaces. Note that the Driver code is written by end-users, and so the Results class must implement the interface, or else the source code for Results will have to be exposed to the end-user.
* Include javadoc style documentation. It is acceptable for this assignment to just have the return type described for each method's documentation.
* For the classes provided in the code template, add interfaces as appropriate

**Design Requirements**

**Late Submissions**

* Same as Assignment-1.

**Grading Guidelines**

Grading guidelines have been posted [here](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/assignments/assign3/assign3Grading.html).

*mgovinda at cs dot binghamton dot edu*

Back to [CSX42: Programming Design Patterns](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/index.html)