**Assignment 5**

**Due date**

* ~~11.59 PM May 9th.~~ 10.00 AM on May 11th.

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**

* Moved build.xml inside src/ folder

**Assignment Goal**

Comparing Objects in Java.

 You are required to use ANT for the following:

* Compiling the code
* running the code
* Generating a tarball for submission

 Your code should compile and run on *remote.cs.binghamton.edu*or *bingsuns* or the *debian-pods*in the Computer Science lab in the Engineering Building.

**Project Description**

**Java Reflection and Object Comparison in Java**

* Extend assignment-4 to use java reflection on instances of First and Second to generate the serialized format following the format used in Assignment-4.
* We will use an input file in which the fields are not missing, and are always in the same order (the order in which the fields are declared in First.java and Second.java).
* First.java and Second.java should be modified to have the fields to match the input files posted for this assignment. First.java has 9 fields and Second.java has 7 fields.
* Keep the code generalized, so that it does not harcode anything about First.java and Second.java. You cannot have code that checks if a reference is an instance of First or Second. Assume your code is generalized if it will work for hundreds of different objects. Use reflection to determine the list of fields of a reference, the type, value, etc. Then, apply the getX(...) methods, where X is the field name.
* Use reflection to set the field values of First and Second instances. So, you need to create the method name (set + fieldName), and setup the method invocation similar to assignment-4. If you use setAccessible(true) and directly set the fields, it violate encapsulation.
* Design a Strategy interface and the corresponding Strategy. The Strategy implementation, DPML, should result in the serializing the objects to their file representation.
* Your code should work when run in the following way. So, do NOT use hardcoded arguments in the build.xml file.
* ant run -Darg0=input.txt -Darg1=output.txt -Darg2=0

**Sample Inputs**

 From YuHeng. Thanks!

* [input-1](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/assignments/assign5/input-1.txt)
* [input-2](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/assignments/assign5/input-2.txt)
* [long-input-10000.txt](https://drive.google.com/a/binghamton.edu/file/d/0B3XP0CblzSkEaWJBZTFibnpIUmM/view?usp=drive_web)
* [long-input-100000.txt](https://drive.google.com/a/binghamton.edu/file/d/0B3XP0CblzSkENnc2UzFxVEx4SWc/view?usp=drive_web)

 From Alex Strong. Thanks!

* [input-3](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/assignments/assign5/input-3.txt)
* [input-4](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/assignments/assign5/input-4.txt)
* [input-5](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/assignments/assign5/input-5.txt)

**Design Requirements**

* Same as previous assignments, except that javadoc is now optional.

**Code Organization**

* Your directory structure should be the following:
* lastName\_firstName\_assign5
* ---genericSerDeser
* ----- README.txt
* ----- input.txt
* ----- src
* -----build.xml
* -----genericSerDeser
* ----------driver
* ----------Driver
* ----------util
* ----------First.java
* ----------Second.java
* ----------PopulateObjects.java
* ----------DPML.java [Implements SerStrategy]
* ----------strategy
* ----------SerStrategy.java [Interface]
* ----------fileOperations
* ----------FileProcessor.java
* ----------other packages that you need
* [package and class(es) to use reflection to create objects

**Code Templates**

* None provided for this assignment.

**Submission**

* Same as Assignment-1.

**Late Submissions**

* The policy for late submissions is that you will lose 10% of the grade for each day that your submission is delayed.

**Grading Guidelines**

[Grading Guidelines](http://www.cs.binghamton.edu/~mgovinda/courses/csx42/assignments/assign5/assign5Grading.html).

*mgovinda at cs dot binghamton dot edu*

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