1) **False Statements:** 1, 7, 8

   Note* : 8 is false because other users can simply modify the .gitignore file. The first part of the statement is correct though.

2)
   a) Inverse Problems:
      i) Integrals
      ii) Determining the original shape of a substance from before a phase change from solid to liquid occurred, using the liquid
      iii) Finding the component values from a resulting sum
   b) Forward Problems:
      i) Derivatives
      ii) Determining the final shape of a substance after a phase change from solid to liquid, using the solid
      iii) Finding the sum of a set of numbers
   c) Complexities:
      i) For integrals it depends. Many can be solved analytically, many can be solved numerically given certain information about the problem, and some cannot be solved analytically or numerically. Analytically solvable integrals have the same computational complexity as it takes to do the math to solve them. For numerical integrals, computational complexity depends on the problem.
      ii) Determining the shape of a solid substance from the shape of the same substance as a liquid after a phase change is impossible due to the loss of information; a liquid substance could have been formed from an infinite number of different solid shapes.
      iii) Finding the component values from a resulting sum is also impossible, unless there are sufficient constraints on the numbers of constituent values and the types of values that were used. The computational complexity of finding the constituent values of the sum would depend on the given constraints and if said constraints make the problem solvable.
      iv) The computational complexity of finding derivatives is similar to integrals in that many are solvable analytically, some numerically, and solve are not solvable, and if differential equations are included, then there are some which are not solvable using either method.
      v) Determining the final shape of a substance from solid to liquid is possible given sufficient information about the initial state and system.
      vi) The computational complexity of finding the sum of a set of numbers is linear with the set's cardinality.

3)
   a) Setting up the Github Repo: SEE GITHUB PAGE @ https://github.com/NightHydra/CognitiveModelingHomework
   b) See Github Repo for merge conflict. Conflict created w/ commit 5a0671ace2f6fc9a6fdc2859e330ba585234256c and resolved in the following

commit. This merge conflict was created via reverting a previous commit. The commit can be found at the link https://github.com/NightHydra/CognitiveModelingHomework/commit/5a0671ace2f6fc9a6fdc2859e330ba585234256c

c) Git command differences
   i) Git restore undos local changes made to files that have not been committed yet. Staged files will stay staged unless the option "--staged" is used in the command. Additionally, git restore undos all changes made locally. Using the staged option means that the changes are not reverted locally.
   ii) Git checkout changes the commit that the files are pointing to. It is used for looking at old commits, however changes should not be made on these prior commits.
   iii) Git reset reverts prior commits and returns the HEAD to a specific number of commits away. Using the "--hard" option means local history will be reverted while the "--soft" option keeps all local changes but makes them unstaged. The –soft option is useful if you dont want a specific commit being in the history due to one simple error which you could easily go back and fix.
   iv) Git revert is used to undo the changes of a specific commit within the history. In order to use this command the head must not have any uncommitted changes. This will also undo any changes made locally even if they are unstaged.

d) See Below for the Filled-In Table

Command	Affects Commit History?	Affects Staging Area?	Affects Working Directory?	Typical Use Case
git reset	yes	no	yes	Undoing commits that are very problematic and difficult to undo manually.
git restore	no	yes	yes	When you accidentally stage a file that you did not mean to stage and want to undo it. Or if you are just playing around with a file and want to go back to the way it was before you started making random edits on it.
git rm	no	yes	yes	This command is used for removing a file and staging that change all in one command. A case where this is useful is with the –cached argument which can remove the file from being

				controlled after it has been added to a gitignore.

4) SEE REPO FOR Python Code
5) See Github Repo For Sample Repo @ https://github.com/NightHydra/CognitiveModelingSampleRepo

I