

**Part I - (Out of Class) - DUE: Friday December 12, 2008**

Your answers for this part of the test are due and are to be "submitted" by midnight Friday December 12, 2008. The relevant "Submit/Review" link on the "course web page" must be used to send your work to me .

Since your responses are to be formulated outside of class, I expect them to be prepared using a "word processor" and to be placed into a single file. I will use MS Word (Office 2007 version) to read the file you submit and I require that all material (tables, diagrams, and any other such content) be embedded in the single electronic document. I encourage you to use a software tool to produce any diagrams.

This portion of the test is not to be regarded as a "group test"; each of you is thus "on your honor" to submit the product of **your efforts** as your work on this portion of the test. This being said, I do not require that you each work in isolation, and do see potential value in some collaboration. Discussing things collectively is good, but when it comes to formulating actual answers I expect each of you to do that individually. If you have any questions please don't hesitate to contact me, and I will do my best to respond with a clarification to the entire class.

1. For the paper, Notation and Representation in Collaborative Object-Oriented Design: An Observational Study, compose a one page summary and critique. For the "summary part" attempt to express the author's essential thesis in your own words, and for the "critique" attempt to express your observations and perspective on this paper. (15%/15%)
2. This question has to do with the Java Program Component for Automatic Singular/Plural Differentiation we have worked on this semester. It is my intent that in answering this question you utilize my solution components that were posted on December 2<sup>nd</sup>, however I am open to the possibility that you may prefer to utilize your submitted components instead. Do **either** A or B below. (15%/30%)
  - A. Apply any fluency you may have in some natural language other than English to develop a "rules file" for that language. Then
    - i. Prepare an annotated "illustration" that demonstrates the applicability of the software for this language. Please understand that I am not fluent in any language other than English (and even that is debatable), so I ask that you provide ample annotations (in English please).
    - ii. Although your linguistic background may be limited, prepare an evaluation of the components ability to serve its intended purpose for natural languages. Offer whatever suggestions/recommendations you think may be warranted.
  - B. Analyze the suitability of the component to support singular/plural differentiation for words other than nouns. Consider pronouns (such as I/we, you/you, etc.), and for adjective and verb agreement (such as is/are). (The wiki article at [http://en.wikipedia.org/wiki/Grammatical\\_number](http://en.wikipedia.org/wiki/Grammatical_number) provides a reasonable preliminary overview on the topic.) What suggestions/recommendations can you offer for extensions or modifications?
3. The text, A Handbook of Software and Systems Engineering: Empirical Observations, Laws and Theories by Endres and Rombach (a link to which is on the course web page) presents a collection of (mostly) useful statements relevant to Software Engineering. The statements of these have been gathered together and may be read off of the course web page at [http://www.cs.scranton.edu/~jackowit/fall2008/se501/SelectedLaws\\_20061103.pdf](http://www.cs.scranton.edu/~jackowit/fall2008/se501/SelectedLaws_20061103.pdf). Pick one law (hypothesis, axiom, etc.) from each chapter that in some way catches your attention and develop a short (about two paragraph) elaboration on and evaluation of your understanding of it. I do not think it necessary for you to have access to the text to answer this question; in fact, I'd prefer you not reference the text. Although the text presents in-depth discussions on each statement, I'd prefer, that for the purpose of this question, you take each statement "as is" and develop then express you own understanding. (15%/45%)

4. In past semesters we have accumulated terms (generally concepts) and worked on developing definitions of each by researching them in textbooks (primary and alternate). The latest manifestation of this "glossary" is available off of the course web page at [http://www.cs.scranton.edu/~jackowit/fall2008/se501/Glossary\\_20060831.doc](http://www.cs.scranton.edu/~jackowit/fall2008/se501/Glossary_20060831.doc), and presents a total of 81 terms without their definitions. For this question you are to do each of the following.
- Analyze the given list for completeness and add terms you think need to be added. The intent here is that the resultant list represents your glossary of the most important terms in Software Engineering. Each term added is to be placed into the numbered alphabetical ordering using a **blue** color so as to differentiate them from the given terms. As you make changes to the glossary (see combinations and removals below), you are to keep the list in numbered alphabetical order and the total number of terms is not to exceed 100. (10%/55%)
  - Combine and/or remove terms currently in the list that you think should be. Indicate these changes by changing the color of each affected term to a **green** color. Each removed term is to remain in the list at its proper alphabetical position but with it's number removed, and each combined term is to remain in the list at its proper alphabetical position with it's corresponding number. (5%/60%)
  - Finally, select any 10 terms from the resulting glossary. The intent is that you will select those terms that you regard as being of significant importance; at least, arguably more important in some way that the terms you have not selected. For each term selected provide a concise definition of no more than a few sentences in length. Present these terms and their definitions as a list.

The substance of your definitions is expected to come from our primary and your alternate textbook and possibly from other sources. So, prepare a color based legend and use the corresponding color coding to identify the source of the specific ideas expressed in each definition. Consider the following legend and example definition. (20%/80%)

**LEGEND:**

Essential of Software Engineering, Tsui and Karam, Jones and Bartlett Publishers, 2007.

Software Engineering, A Practitioner's Approach - 6<sup>th</sup> Ed, Pressman, McGraw-Hill, 2005.

Wikipedia, [http://en.wikipedia.org/wiki/Software\\_engineering](http://en.wikipedia.org/wiki/Software_engineering), December 2008.

Software Engineering - A broad field that touches upon all aspects of developing and supporting a software system, through the establishment and use of sound engineering principles in order to economically obtain software that is reliable and works efficiently on real machines. Software engineering is related to the disciplines of computer science, project management, and systems engineering.

5. This question pertains of Chapter 10 - Testing and Quality Assurance of the Tsui and Karam text. Do each of the following.
- Using the technique of "equivalence class partitioning" develop a "test set"(a collection of "test cases") to be used to verify the Java Program Component for Automatic Singular/Plural Differentiation. Be sure to clarify your answer. (10%/90%)
  - Use the technique of "path analysis" to develop a "test set" to verify the method `addWord` in the `WordTree` class provided at

[http://www.cs.scranton.edu/~jackowit/fall2008/se501/AS03R/PMJ\\_Solution/WordTree.java](http://www.cs.scranton.edu/~jackowit/fall2008/se501/AS03R/PMJ_Solution/WordTree.java). Note that this is the specific method we considered and conjectured about in class on Thursday Dec 4<sup>th</sup>. Note that as now given the questionable section is "commented out" and it still appears to work. The motivation for this analysis is our desire to verify just this specific method. (10%/100%)

Good luck,  
P.M.J. - *December 5, 2008*