CSC 276 Rubric Contract Grading: Design

The following rubric is used to assess the level of student learning in CSC 276 as it relates to creating a software design. See the class syllabus for details on how this rubric is used. For purposes of computing an assignment grade, any criteria deemed unacceptable shall have a numeric value of (50%). All criteria are weighted equally when averaging for an assignment grade.

Criteria	Acceptable (C=75%)	Better (B=85%)	Best (A=100%)
Modeling techniques	• Two or more notations are not valid (not part of the modeling technique). • Two or more notations not used properly.	• One notation is not valid (not part of the modeling technique). • One notation not used properly.	• All notations on a diagram come from same modeling technique. • All notations on a diagram are used properly based on the semantics of the modeling technique.
Design artifacts	• Two major inconsistencies in the design artifacts (or one major inconsistency and lots of small inconsistencies).	• One major inconsistency in the design artifacts (or lots of small inconsistencies).	• All design artifacts are consistent with each other (or have a few small inconsistencies).
Model-View-Controller	• Two major errors in how MVC is described in the design artifacts. • Two major flaws in how MVC is implemented in code.	• One major error in how MVC is described in the design artifacts. • One major flaw in how MVC is implemented in code.	• All three components are described correctly in design artifacts. • All three components are implemented correctly in code.
Design elements	• Architecture description is not clear/concise, not clear/complete, xor not concise/complete. • Data description is not clear/concise, not clear/complete, xor not concise/complete. • Interface description is not clear/concise, not clear/complete, xor not concise/complete. • Components description not clear/concise, not clear/complete, xor not concise/complete.	• Architecture description is not clear, not concise, xor not complete. • Data description is not clear, not concise, xor not complete. • Interface description is not clear, not concise, xor not complete. • Components description not clear, not concise, xor not complete.	• Architecture description is clear, concise, and complete. • Data description is clear, concise, and complete. • Interface description is clear, concise, and complete. • Components description is clear, concise, and complete.
Design Criteria	• Simplicity: One major (or many minor) adjustment(s) to the design would make the design simpler. • Coupling: One major (or many minor) adjustment(s) to the design would result in lower coupling. • Cohesion: One major (or many minor) adjustment(s) to the design would result in higher cohesion. • Information hiding: One major (or many minor) adjustment(s) to the design will improve information hiding. • Performance: One major (or many minor) adjustment(s) to the design will improve the performance. • Security: One major (or many minor) adjustment(s) to the design will improve the security of your app.	• Simplicity: A few minor adjustments to the design would make the design simpler. • Coupling: A few minor adjustments to the design would result in lower coupling. • Cohesion: A few minor adjustments to the design would result in higher cohesion. • Information hiding: A few minor adjustments to the design will improve information hiding. • Performance: A few minor adjustments to the design will improve the performance. • Security: A few minor adjustments to the design will improve the security of your app.	• Simplicity: Zero or one minor adjustment to the design would make the design simpler. • Coupling: Zero or one minor adjustment to the design would result in lower coupling. • Cohesion: Zero or one minor adjustment to the design would result in higher cohesion. • Information hiding: Zero or one minor adjustment to the design will improve information hiding. • Performance: Zero or one minor adjustment to the design will improve the performance. • Security: Zero or one minor adjustment to the design will improve the security of your app.
Implementation	• Two major inconsistencies between the design artifacts/elements and the code.	• One major inconsistency between the design artifacts/elements and the code.	• All design artifacts/elements are correctly implemented in the code.