CS 246 Project Grading Rubric

		Exceptional 100%	Good 90%	Mostly Adequate 80%	Developing 70%	Minimal 50%
Core Functionality	How well does the project work? – 35%	Code compiles, runs with no exceptions, and has excellent test cases that exercise the business logic.	Code compiles and runs with zero run time exceptions, and has some test cases.	Code compiles and runs fairly well under normal circumstances, but may break down in unusual circumstances, or is lacking test cases.	Code compiles, and resembles the solution, but has several bugs that occur during normal circumstances.	Some resemblance to the solution exists, but it is very minimal.
	How well does the project fulfill its requirements? – 25%	All explicit requirements met, and some stretch ones too.	All explicit requirements met.	The majority of requirements met, but still lacking needed functionality.	About half of the requirements met.	Fewer than half met.
Code Style, Design, and Development	How well does the code embody good design principles and patterns? – 10%	Exceptional use of appropriate design patterns *AND* OO design principles and practices (e.g., interfaces, inheritance, composition, etc.).	Good application of OO design principles and practices (e.g., interfaces, inheritance, composition, etc.).	Many principles of good OO design are present, but the project lacks proper overall design and cohesion.	Some elements of good OO design are present, but almost as an exception.	Code is not simply in one big class.
	How "beautiful" is the code? (e.g. conforms to Java code conventions) – 10%	Perfect compliance with course code standards document, and excellent naming.	Nearly consistent, but some areas of formatting and/or naming could be cleaned up.	Most of the code looks good, but there are a number of areas of bad naming or very poor formatting.	Some of the code is consistent with the standards, but it is more an exception.	An attempt was made at correct conventions.
	How well documented is the code? – 10%	Someone else could easily dive into my project next semester based solely on the comments and the Javadoc headers.	Javadoc headers blocks are filled in consistently and some comments exist.	Header blocks are mostly present.	Some comments and Javadoc headers exist, but it is not consistent.	Some comments here and there.
	How well was version control used? – 10 %	Specific targeted changes committed with excellent comments.	Specific targeted changes committed, but comments should be improved.	Decent commits, but should have been more frequent, and comments are not very helpful.	Changes committed, but caused build errors, or too much at once, and terrible comments.	Only committed occasional huge changes.