Computer Programming Grading Rubric

	Excellent (5)	Proficient (4)	Satisfactory (3.5)	Novice (2)	Unsatisfactory (0)	Score/Level
	Student met learning objective. Demonstrated 100% mastery of the concept.	Student met learning objective. Demonstrated 90% mastery of the concept.	Student is approaching learning objective. Demonstrated 80% mastery of the concept.	Student has not met learning objective. Demonstrated under 70% mastery of the concept.	Student has not met learning objective. Provides little to no evident understanding of concept.	/100
Program Design (pseudocode, Flowchart, documentation)	Ex. Pseudocode clear, brief and well organized	Ex. Clear, understandable but steps too detailed	Ex. Somewhat understandable but does not complete	Ex. Difficult to understand and/or unorganized	Ex.no pseudocode or/and chart	/ 10(pseudocode only)
Overall functionality of code	Ex. Code organized and delivers accurate results	Ex. Code is good, delivers results	Ex. Code is adequate, but did not meet all requirements.	Ex. Code is inadequate, generates logical errors	Ex. Code is missing 80% of requirements or cannot run	/20
• **Branching** (decision structure) • **Variable use and naming** • **User input/ Output**	Ex. Efficient use of skills (all possible cases handled)	Ex. Proficient use of skills (mostly complete)	Ex. Adequate use of skills	Ex. Some of the skills not utilized properly	Ex. Few or none of the skills utilized	/30
Proper use of loops • **For loops** • **While loops**	Ex. Efficient use of skills	Ex. Proficient use of skills	Ex. Adequate use of skills	Ex. Some of the skills not utilized properly (for example use of *while True and Break*	Ex. Few or none of the skills utilized	/30

				and continue)		
Functions and Modular programming	*Ex. Exceptional Proper use of modularization, required functions created with well-defined Docstrings*	*Ex. Proficient. Code modularized, Functions created, some essential info missing from Docstring*	*Ex. Code exhibits 70% of requirements (Ex no Docstring, long code not modularized...)*	*Ex. Code not modularized if needed, or/and some functions missing*	*Ex. Few or none of the skills utilized*	
Object Oriented Programming	*Ex. Classes /Inheritance perfectly implemented, with private attributes, docstrings and necessary methods*	*Ex. Classes implemented, necessary methods created, attributes private. Inheritance partially implemented*	*Ex. Classes/inheritance partially implemented, demonstrating limited understanding*	*Ex. Poor implementation of skills (Ex. Inheritance not implemented, methods missing...)*	*Ex. Few or none of the skills utilized*	
File I/O, String, Exception Handling	*Ex. Efficient use of skills (all requirements met and best practices applied)*	*Ex. Proficient (requirements met, efficient use of skills could be better demonstrated)*	*Ex. 70% of requirements met. Limited understanding (Ex use of learned skills not applied)*	*Ex. Code not addressing all requirements*	*Ex. Few or none of the skills utilized*	
Collections/ Data Structures	*Ex. Efficient and correct use of Data Structures*	*Ex. Proficient (Requirements met but some skills not applied)*	*Ex. 70% of requirements met. Limited understanding (Ex use of learned skills not applied)*	*Ex. Code does not meet all requirements, inaccurate choice of structures*	*Ex. Few or none of the skills utilized*	
User Interface	*Ex. Efficient, clear, well formatted, data validation applied*	*Ex. Proficient (some aspects of the interface can be enhanced)*	*Ex. 70% of requirements met*	*Ex. Interface minimally meet specifications, unpresentable and not easy to navigate*	*Ex. Interface incomplete, does not meet specifications*	
Other Coding Standards	*Ex. Code adheres to best standards and uses concepts referenced in*	*Ex. Code utilizes concepts referenced in the course, some best practices not*	*Concepts/standards learned in class not fully applied*	*Limited demonstration of concepts/standards learned in class*	*No utilizations of concepts and best practices used*	/10

	course	applied.				