

RUBRIC ASSIGNMENT 3: ROMAN NUMERAL CONVERTER

Criterion	Good		Satisfactory		Poor		Not Met	
Coding Style	Code is clean, understandable, and well-organized. Good use of indentation, whitespaces, paragraphing and variable names.	2	Minor issues with consistent indentation, use of whitespace, variable naming, or general organization.	1	At least one major issue with indentation, whitespace, variable names, or organization.	0.5	Major problems with at three or four of the readability subcategories.	0
Documentation	Comments are well written and clearly explain the code. Contains a program header.	2	One or two places that could benefit from comments are missing them or the code is overly commented.	1	Program header missing, complicated lines or sections of code uncommented or lacking meaningful comments.	0.5	No program header or comments present.	0
Usability	Program is intuitive to use, provides user feedback.	2	Program is somewhat intuitive to use, provides some user feedback.	1	The program is confusing to use with a lack of feedback.	0.5	Usability is extremely poor.	0
Reflection report	Clearly describes the Cobol design process.	6	Somewhat describes the Cobol design process.	4	Describes the Cobol design process in a very limited manner.	2	No reflection report.	0

Not Met - Criteria not met.

Poor - Evidence that the student has mastered this learning objective is not provided, unconvincing or incomplete.

Satisfactory - Evidence shows that the student has generally attained this objective. Some of the program structures remain in their legacy state.

Good - Evidence demonstrates that the student has mastered this objective.

Run-time	Program has serious run-time issues and crashes.	-4	Program has some run-time issues and crashes on occasion.	-2	Program has no run-time issues and always functions.	0	Program <u>does not compile</u> or code <u>does not differ greatly</u> from the supplied legacy code. Assignment receives a grade of: 0
Compiler warnings	Program produces >4 compiler warnings.	-2	Program produces ≤4 compiler warnings.	-1	Program produces no compiler warnings.	0	

GRADE	OVERALL <u>35</u> / 40	LATE DEDUCTION: (1 / 2 days) × 0.9 / 0.8	=	FINAL GRADE 35 / 40
--------------	---------------------------	--	---	-----------------------------------

35

RUBRIC ASSIGNMENT 3: ROMAN NUMERAL CONVERTER

		Good	Satisfactory	Poor
	Identifiers used in the program have better readability. e.g. NO-OF-SENTENCES to numSent	1.5	1	0.5
	Structural statements have been modified to use explicit terminators, e.g. end-if, end-read	1.5	1	0.5
Re-engineering	Reduce "wordiness" of program, i.e. DIVIDE X INTO Y GIVING Z converted to COMPUTE Z=X/Y	1.5	1	0.5
	The programs iterative structures, e.g. loops, calling paragraphs, have been rationalized	1.5	1	0.5
	The general structure of the program has been made more readable e.g. paragraphs	1.5	1	0.5
	GO TO statements have been removed.	4.5	3	1.5

Correctness	Program always works correctly. i.e. all test cases are correct.	6	Program is partially correct. i.e. all test cases produce results which are partially incorrect.	4	Some test cases fail completely. e.g. one passes, the other fails.	2	Program does not function at all - may contain logic errors.	0
--------------------	---	---	--	---	---	---	--	---

	Good	Satisfactory	Poor
Input	The program allows the Roman numeral to be input in a linear fashion, e.g. XVI 5	The program allows the Roman numeral to be input in a linear fashion, with no instructions. 3	The program performs input in the original manner. 1
File I/O	The program allows file input. 3		No file input 0
BEYOND (5%)	The program takes both upper and lower case, and validates input. 2	The performs only one of (i) upper and lowercase, or (ii) validating input 1	Neither activity performed. 0