Explanation:

This grading rubric has been created to provide a consistent means for peer-evaluations by giving detailed assessment explanations and scoring guidelines. This is to provide all peer-assessors a means for assessing assignments in a consistent manner and help in providing meaningful feedback. This not only helps those doing assessments, by explaining what they should look for in assessing each criteria. It also assists those who submitted the assignment being assessed, but enabling them to know the exact criteria and scoring guidelines that were used, along with the assessor's feedback, to explain why they obtained each score they did.

This rubric is provided as a PDF (You're welcome, but NOT required, to print it out and use it as a worksheet)

Instructions:

There are several Evaluation Categories (Rows with grey backgrounds) in this rubric. Each Category has one-or-more specific Criteria(an individual row) for you to assess. There are 4 possible columns in the rubric which represent the Assignment's level of completion for each Criteria. Each box has written explanation of it and a scoring guide in the bottom right of it. However, some criteria (and therefore that row) won't have 4 valid unique potential assessment groupings. Therefore, that row will have boxes that take up more than one column.

Please read each Criteria(row) being evaluated, and then read each Box's explanation. Find the Box that best describes the state of the assignment being assessed. Determine the score for that Criteria(row) based on the scoring guideline in the bottom right of that matching Box. For each criteria(row) you will need to keep track of the points awarded. You will have to enter these values into the platform along with a short explanation explaining why you chose the score you did.

	Exceptional	Acceptable	Beginner	Unsatisfactory	Points Earned
Assignment Delivery					
Assignment zip file Contains all required files and compiles	The assignment zip file contains the App and all the required files. You were able to import the assignment properly into Android Studio. 10 / 10			The zip file is missing files, or has syntax errors when imported. 0 / 10	___ / 10
Functional Requirements					
Button Press Results in Output (output doesn't have to be correct for this part)	Each and every button press results in an updated value being displayed in the results area 5 / 5		Button press works most of the time, but doesn't update properly each press or is otherwise unreliable. 2 / 5	Button press doesn't result in any output at all 0 / 5	___ / 5
'Add' works	The 'Add' Functionality works properly, You tested it with multiple values and it properly calculated each pair. 15 / 15		Add works for some number pairs but not all, such as when one or two arguments are negative. 5 / 15	'Add' doesn't work at all, either there is no output or the output is always wrong. 0 / 15	___ / 15
'Subtract' works	The 'Subtract' Functionality works properly, You tested it with multiple values and it properly calculated each pair. 15 / 15		'Subtract' works for some number pairs but not all, such as when one or two arguments are negative. 5 / 15	'Subtract' doesn't work at all, either there is no output or the output is always wrong. 0 / 15	___ / 15

Functional Requirements cont.					
	Exceptional	**Acceptable**	**Beginner**	**Unsatisfactory**	**Points Earned**
‘Multiply’ works	The ‘Multiply’ Functionality works properly, You tested it with multiple values and it properly calculated each pair. 15 / 15		‘Multiply’ works for some number pairs but not all, such as when one or two arguments are negative. 5 / 15	‘Multiply’ doesn’t work at all, either there is no output or the output is always wrong. 0 / 15	___ / 15
‘Divide’ works	The ‘Divide’ Functionality works properly, You tested it with multiple values and it properly calculated each pair. 15 / 15		‘Divide’ works for some number pairs but not all, such as when one or two arguments are negative, or division by 0 was allowed and didn’t give error feedback. 5 / 15	‘Divide’ doesn’t work at all, either there is no output or the output is always wrong, or divide by 0 wasn’t handled properly with an error message. 0 / 15	___ / 15
Object Oriented(OO) Design Quality					
Made good use of each of the Add, Subtract, Multiply, & Divide classes	Each of the 4 math-operation classes followed good OO Design, meaning they took in both argument values in one of three ways: 1) In constructor parameters, 2) ‘setter’ methods for each value or 3) as parameters to a commonly named method between the 4, such as ‘operate(int arg1, int arg2)’ 5 / 5		The 4 operations classes had static methods that did the calculations for each operation, or they did not use the 4 calculations classes at all. 0 / 5		___ / 5
Added some other Good Object Oriented design	This section is very broad and can have a -LOT- of potential answers. Most likely this would be making use of an Interface that each of the 4 math-operation classes would implement (such as ‘operate(int arg1,int arg2)’, or having a base class (abstract or not) that would have a unifying method that all 4 classes would call. The use of generics would also fulfil this requirement. Especially in this section, be generous, if their explain (in documentation) seems reasonable, even if you wouldn’t do it that way, then error on the side of too many points 5/5			They didn’t use any Object Oriented design that you can see, and they didn’t document which principle, pattern, or language feature they made use of. Or their documentation for what ‘Obj. Oriented’-Design isn’t what they made. 0/5	___ / 5

Source Code Aesthetics

	Exceptional	Acceptable	Beginner	Unsatisfactory	Points Earned
Comments	The code is well-documented with appropriate names and comments that explain what the code is accomplishing and how. 5 / 5		The code has very little documentation, such as: classes, method, variables or even sets of variables, and tricky pieces of code are not documented. 0 / 5		___ / 5
Indenting and spacing	The author of the code used a consistent indenting and spacing pattern throughout their code (the specifics of -exactly- what pattern of indenting and spacing, does not have to be 'exactly' what you would use, or what was talked about in the course.) Consistency is more important than the specifics of what pattern they used, so as long as they were consistent throughout most of the assignment. Please error on the side of giving credit/points. 5 / 5		The author of the code was consistent with indenting and spacing over half of the time, but had a couple areas that weren't and those sections of code distracted from the readability of the code. 2 / 5	The author of the code was not consistent with indenting and spacing in their code. 0 / 5	___ / 5
Class, Variable, and Method naming	Class(es), Variable(s), and Method(s) that were added to the assignment are meaningful and follow the Java guidelines for Capitalization Pascal-case for Class Names (All words Upper-case, including first word), snake-case for methods and variables(All words Upper-case, -except- first word). 5 / 5		Class(es), Variable(s), and Method(s) that were added to the assignment do not have meaningful names and are not capitalized in the standard way. 0 / 5		___ / 5