



Activity: Conditional Statements

OVERVIEW:

For this assignment, you will create a calculator that collects information from the user with prompts, calculates some sort of useful result from that information, and prints the result to the console. You will create code for dynamic decision-making with conditional statements. You will also be responsible for creating a flowchart that reflects the logic of the code you create.

LEVEL OF EFFORT:

This activity should take approximately 240m to complete. It will require:

- 0m Research
- 15m Prep & Delivery
- 225m Work

If you find that this activity takes you significantly less or more time than this estimate, please contact me for guidance.

READING & RESOURCES:

Instructional Conditional Statements - Rubric (necessary)

This rubric outlines the points for the assignment. Make sure you check off each one as done before submitting your assignment.

OBJECTIVES:

Successful completion of this activity will show that you can do the following:

- Determine when to use if, else, and else if conditional statements.
- Formulate conditional statements for dynamic decision-making.
- Employ logical operators to combine conditions.
- Employ relational operators to weigh conditions as true or false.
- Create logic to resolve a single solution from a selection of possible solutions.

INSTRUCTIONS:

1. Before you begin, you should read the rubric. This is extremely important as it will tell you exactly how this assignment will be graded.
2. In your class folder, within your repository, create a folder called **Lastname_Firstname_Conditionals_Assignment**.
3. Brainstorm an original idea for a calculator of some kind. This should not be anything we have covered in class and not the same as any of your other projects.
Keep in mind that your final project must contain the following:
 - a. An example of 2 different arithmetic operators (addition *and* subtraction, addition *and* multiplication, subtraction *and* division, etc.).
 - b. 2 conditional statements – A least one must contain an else if statement
 - c. At least 1 ternary conditional statement must be used
 - d. At least 1 Logical Operator && , || or !
 - e. Must contain at least 3 user prompts that are used in calculations
 - f. All prompts must be validated
4. Using the flowcharting skills learned in Homework #1, create a detailed flowchart for your process. Save this as a PDF into your Conditionals Assignment folder and commit it to your GitHub Repo. Name this file **Lastname_Firstname_Conditionals_Flowcart.pdf**
5. Once your flowchart is created, make sure you follow the organization mentioned in the videos to create a html, js folder and js file inside of the Conditionals_Assignment folder.
 - a. Place your name, date, and assignment at the top of each of the JavaScript files in comments.
 - b. Link the js file to the HTML page, and verify it runs correctly.
 - c. Following your flowchart, write your code to match. Make sure to comment every line so that you are explaining exactly what you are trying to do.
 - d. Continuously commit your files to your GitHub repo throughout the development process. You should have a minimum of 5 meaningful commits for this assignment.
 - e. Your code should give the user meaningful output. So, after your calculations are complete, your code should report back to the user the final values with a console.log(). This should contain the variables that you calculated and a concatenation text string that describes the value.
e.g. `console.log("The area of the rectangle is " + calcArea + "!");`
 - f. After your code is complete, make sure you test your final files. Try typing in different values to see what you get. Put these test values in a multi-lined comment at the bottom of your js file.
e.g. `/* I typed in 5 for length and 6 for height and my calculator gave me 30 for the area. */`
6. Please note that you are not required to do your flowchart before your code. That is, you may complete your code and then go back and create your flowchart. You may also choose to create your flowchart at the same time you're creating your code. The important thing to remember is that logic of your flowchart must match the logic of your code and vice versa.

CRITERIA:

Make sure your project follows this list of criteria:

- You will need to create code that calculates a solution based on the values entered by a user. The actual values don't matter; you should be able to type in any numbers in the prompts, which change the values of your variables, and the calculation should still be accurate.
- These user prompts must be Validated. This is **IMPORTANT** as it is 10% of your grade!!
 - This means your code should check whether or not the user left a prompt entry *empty*.
 - For example, if the user does not enter a value for the first prompt, make a conditional that checks if the value is empty. You can do this by evaluating if the contents of the prompt are the same as an empty string (which looks like "" i.e. that's two quotations with no space in between them)
 - Here is an example of some possible pseudocode (english representing the logic in the code you write)
 - //Collect info from prompt1 and place it into a variable
 - //If the variable is an empty string
 - //Put another prompt saying "You forgot to input something!"
 - //Collect that info from that prompt and place it into a variable
 - Otherwise
 - //Continue on with the rest of the code...
- Decide on what your code will calculate. This should be an **original** calculation. In other words, you may not use any of the examples mentioned in this class. This will result in an automatic **0**.
- Your code must contain a minimum of two conditional statements
 - 1 conditional statement must contain an else if statement
- Your code must contain at least 1 ternary conditional statement
- Your code must contain at least one Logical operator ie. &&, || or !
- There should be a minimum of **3 prompts** to collect user information. The prompts should include labels so the user knows what information they need to enter. These prompts should assign the information collected into variables you have set up.
- Variable names should be descriptive of the values they are assigned.
- Calculate the result using the *variables* in an expression (NOT literal values or numbers).
- Assign the calculation result to the a variable.
- The result should appear in the browser's console and include an explanation of the result.
 - **Good example of console print out:** The volume of the sphere is 26 feet cubed.
 - **Bad example of console print out:** 26
- Final output should use string concatenation.
- Comment every line of code (describe what each line is doing in English). Do NOT just label sections of your code.

TURNING IT IN:

- Double-check that you've commented your code (You can't comment too much).
- Commit and push your files to your GitHub repository.
- Compress your **Lastname_Firstname_Conditionals_Assignment** folder into one zipped file. It should be named **Lastname_Firstname_Conditionals_Assignment.zip**
- Upload this zipped file to FSO. NOTE: This file is only used as a backup file. Everything will be graded based on the files pushed to your GitHub repository.
- **IMPORTANT:** If your code is not in your GitHub repository, it will result in a 0 for the assignment.

DON'T FORGET:

- Test your code. Do the calculations on your own calculator, and check that the computer is producing the same result.
- Test each path of every conditional. There should be a difference.
- Test your validation of your prompts.
- This is not a "calculator" in the sense of a device with number buttons. This is meant to collect information from the user, use that information in a very specific way, and deliver a result.
- You should have a minimum of 5 commits in your GitHub repository.

Week 3: Conditionals

Scalable Data Infrastructures: MDV2330

Bare Minimum Requirements

These requirements must be satisfied before any points are awarded. Failing to meet these requirements will result in a zero (0) grade.

1. Working JavaScript file with no major syntax errors.
2. HTML file with working script link to the JS file.

Topic	%	Excellent (100%)	Good (75%)	Fair (30%)	Poor (0%)	
Programming Fundamentals						
Flowchart	10	Flowchart is present and contains: 1. Sequential & Conditional process shown 2. A clear beginning and end 3. Correct Symbols	Missing one of the previous list.	Missing two.	No flowchart was submitted with the project.	
Calculation	5	Some type of math is used an a calculation and includes at least 2 factors as variables, not hard-coded values.	Calculation is used, but only with 1 factor.	Numbers are used, but calculation is not present or meaningful, or hard-coded values are used.	No calculation is present, or is the same as the examples in the course material or the previous assignment. (Zero for the entire project.)	
Conditionals	15	2 Conditional Statements are present in code and work correctly.	Only 1 Conditional Statement is present in code.	Conditionals are not present, not meaningful, or lack of insight is shown. (Zero for the entire project.)		
Else If Statement	5	At least 1 Else If Statement is present in code and works correctly.		No Else If Statement is present.		
Ternary Statement	5	At least 1 Ternary Statement is present in code and works correctly.		No Ternary Statement is present.		
Logicial Operator	5	At least 1 Logical Operator is present in code and works correctly.		No Logical Operator is present.		
User Prompts	5	3 user prompts are correctly used in the code.	2 out of 3 user prompts are correctly used in the code.	1 out of 3 user prompts are correctly used in the code.	No user prompts are present in code.	
Validation	10	All 3 prompted values are checked to make sure they look okay.	2 out of 3 prompted values are checked to make sure they look okay.	1 out of 3 prompted values are checked to make sure they look okay.	Validation is not present or not meaningful.	
Comments	10	Each line of code has a meaningful comment.	Missing a few comments.	Missing several comments, or comments are not meaningful.	No meaningful comments.	
Technical						
GitHub Delivery	10	Files are available on GitHub, visible to the instructor, and in an easily-discoverable location, and include a minimum of 5 meaningful commits.	Files are on GitHub, but not visible to the instructor or hard to find, or fewer than 5 commits.		Files are not on GitHub.	
Files	10	File structure is as specified and student name and date is present in all work.	Missing student name or date, or minor file structure issues.	Major file structure issues.	Files are a hot mess.	
Professionalism						
Completeness	5	All required elements from the assignment are included.	One or two minor elements missing.	Major pieces missing.	Not even close to complete.	
Investment	5	Presentation is well thought out, coherent, and includes personal relevance.		Presentation is impersonal or disconnected.	Little to no forethought, personalization, or relevance.	
Your course Professionalism grade is affected by your Investment grade.						