**PLANNING A COMPLEX ALGORITHM**

**DESIGN THE ROUTINE**

CHECK PREREQUISITES

Define the problem

*We want to add json data reading*

Information the routine will hide

*Calculation and parsing of data.*

Inputs to the routine

*File Choice for data parsing*

Outputs from the routine

*Adds all info to tournament and displays correctly*

Pre-conditions

*Json data file prepared. File loader compatibility in browser*

Post-conditions

*Tournament is set up correctly and data displays the same way*

Name the Routine

*Reading from JSON data file*

Decide how to test the routine

*Console.log will be used to display that the tournament is setup correctly, and that the tables still produce correctly will be a good indicator that the set up will work.*

Research functionality available in standard libraries

*The majority of this assignment is working with HTML DOM, which is readily available within JS Libraries. I will have read json and parse it, aswell as html input elements for file uploads.*

Think about error handling

*Generally, with this dynamic generation Physical errors are able to be identified and fixed as data will not return on the page. Otherwise all errors will be caught by the console. Also some elements will not change if an error occurs halfway through the function execution.*

Think about efficiency

*Looping will be used to generate the majority of the data. This approach will avoid the Hard Coding of every single result.*

Research algorithms & data types

*Dom Elements have a different syntax of assigning variables and attributes than regular html. Assigning them to a variable then using functions to assign these attributes seems to be the simplest way of doing it.*

**WRITE PSEUDOCODE**

1. Think about the data
2. Check the pseudocode
3. Try ideas in pseudocode

**CODE THE ROUTINE**

1. Write the declaration
2. Turn pseudocode into comments
3. Fill in code below comments
4. Check if code can be factored

**CHECK THE CODE**

1. Mentally check for errors
2. Step through in Debugger
3. Test the code
4. Remove errors in the code
5. Clean up