# Independent Study Project – Checkpoint 1

## Purpose

To create a product that engages you and that you would be proud to share to a public audience.

Along the way, you will develop your ability to problem-solve using a variety of strategies, to implement a solution in code, to manage source code using accepted industry practices, and to plan and meet commitments for project milestones.

## Evaluation

As described in January, I am now taking a standards-based approach to evaluating your progress in the course.

What does that mean?

It means that I value the *process* of your work on this ISP as much as your *product.*

It means that I am looking, quite simply, for you to provide evidence of having met the expectations listed.

To that end: using your commits on GitHub, and your posts on Sesame, how would *you* evaluate your progress so far?

You probably will not have yet demonstrated *all* of the expectations, but have you hit some? How often?

For each expectation shown on the following pages:

1. Provide links(s), optionally with brief explanatory text to specific parts of a commit in your source control history
2. Give yourself a 1 to 5 star rating

## Curriculum Expectations

### A1. Data Types and Expressions Demonstrate the ability to use different data types, including one-dimensional arrays, in computer programs;

**A1.1** use constants and variables, including integers, floating points, strings, and Boolean values, correctly in computer programs;

ASCII, Unicode) to internally represent data and store information;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L27-L46 |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A1.3** use assignment statements correctly with both arithmetic and string expressions in computer programs;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L237-L272 |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A1.4** demonstrate the ability to use Boolean operators (e.g., AND, OR, NOT), comparison operators (i.e., equal to, not equal to, greater than, less than, greater than or equal to, less than or equal to), arithmetic operators (e.g., addition, subtraction, multiplication, division, exponentiation, parentheses), and order of operations correctly in computer programs;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L266-L272 |
| https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L233-L234 |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A1.5** describe the structure of one-dimensional arrays and related concepts, including elements, indexes, and bounds;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L97-L125>  This link shows my knowledge of one dimensional arrays because I utilize keys to access certain parts of my JSON data, which is an array. I am able to access certain arrays in the larger array (this is a 2D array) by using “keys,” which are the names of arrays. After I use this key, I am able to access this array, which I can then get the data from that array using another key, which is the name given to the data, such as “home\_team\_name.” This whole system is a dictionary, which holds all of the data that is gathered, and sets them to certain keys so they can be accessed when by just imputing the key and then taking the data that is associated with that key. |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A1.6** write programs that declare, initialize, modify, and access one-dimensional arrays.

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L160-L175 |
| <https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L192-L200> |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

### A2. Control Structures and Simple Algorithms Demonstrate the ability to use control structures and simple algorithms in computer programs;

**A2.1** write programs that incorporate user input, processing, and screen output;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/rsgc-canavan-s/BlueJaysISP/commit/5ccfb651667ebb1c6fbde1d5e8822ca95ada6e46> |
| <https://github.com/rsgc-canavan-s/BlueJaysISP/commit/a3583235cd18af4fcf0bc6d0abab19e337662a41> |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A2.2** use sequence, selection, and repetition control structures to create programming solutions;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L137-L279>  This for loop is what I use to iterate over the array that contains all of the arrays that contain data for the games |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A2.3** write algorithms with nested structures (e.g., to count elements in an array, calculate a total, find highest or lowest value, or perform a linear search).

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
|  |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

### A3. Subprograms Demonstrate the ability to use subprograms within computer programs;

**A3.1** demonstrate the ability to use existing sub-programs (e.g., random number generator, substring, absolute value) within computer programs;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L95> |
| <https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L63>  <https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L54>  <https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L241> |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A3.2** write subprograms (e.g., functions, procedures) that use parameter passing and appropriate variable scope (e.g., local, global), to perform tasks within programs.

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L83> |
| <https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L50> |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

### A4. Code Maintenance Use proper code maintenance techniques and conventions when creating computer programs.

**A4.1** demonstrate the ability to identify and correct syntax, logic, and run-time errors in computer programs;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| I realized that I did not need to use an if let statement, and changed if after I got an error |
| See sesame post from April 28, 2017 |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A4.2** use workplace and professional conventions (e.g., naming, indenting, commenting) correctly to write programs and internal documentation;   
 (also includes use of source control)

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L38-L46> This section of my code is all of the names that I gave to variables, which I believe are very descriptive and pertain to the data they hold. |
| <https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L151> additionally, this is just one of the variable names that I gave to the data that I parsed, but I believe that it is a good representation of my naming of variables |
| <https://github.com/rsgc-canavan-s/BlueJaysISP/blob/master/JaysAppParsingJSON.playground/Contents.swift#L185> I believe that this line of code demonstrates my overall commenting job in my code |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A4.3** demonstrate the ability to interpret error messages displayed by programming tools (e.g., compiler, debugging tool), at different times during the software development process (e.g., writing, compilation, testing);

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| See above in A4.1 |
| See April 28 2017 sesame post |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A4.4** use a tracing technique to understand program flow and to identify and correct logic and run-time errors in computer programs;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
|  |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A4.5** demonstrate the ability to validate a program using a full range of test cases.

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| See sesame post April 28 2017 at 2:12 |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

### B1. Problem-solving Strategies Use a variety of problem-solving strategies to solve different types of problems independently and as part of a team;

**B1.1** use various problem-solving strategies (e.g., stepwise refinement, divide and conquer, working backwards, examples, extreme cases, tables and charts, trial and error) when solving different types of problems;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| I have encountered many different errors, and have tried to fix them with many different methods, mainly trial and error. |
| When I worked with Mr. Gordon, we came up with a chart of what to parse and what type of data that we would need to obtain to get to the next part of the code |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**B1.2** demonstrate the ability to solve problems independently and as part of a team;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| I have been able to deal with many errors on my own, such as the two examples that I gave above and in sesame |
| I have also worked with Mr. Gordon to help start me on parsing my code, just getting started and learning how to obtain data from JSON, as well as getting into the JSON using keys and guard let statements |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

## Comments and Proposal for Level of Achievement

Understanding that this is a checkpoint 1/3 of the way into the ISP, and that mastery of all standards is not expected at this point in time, what do you suggest as your current level of achievement? Why?

I have not done as much in the way of writing code, but I have managed to create the launch screen and the home screen. Because all of the UI stuff is new to me, I required a lot of research and caused plenty of errors which took time to fix. I have found a JSON file that I can parse, which I hope to have parsed for the next checkpoint, although I do have to learn how to use the JSON parsing library. I have also begun to create the sub-screens after you touch one of the buttons on the home screen. For this, more research has to be done because it requires using the UI Tables, something I am unfamiliar with. Because I am doing an app that uses parsed JSON files from MLB.com, I don’t think that I will be doing a lot of the code that my peers who are making games are.