# Software Project I: Algorithms

## Purpose

To create (a) product(s) that engage(s) 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 design an algorithm, to implement a solution in code, to manage source code using accepted industry practices, and to plan and meet commitments for project milestones.

## Evaluation

75% of your final grade on a product comes from your ability to provide regular evidence that you have met curriculum expectations in this course.

Using your final commit GitHub, and your posts on Sesame, what have you learned and demonstrated knowledge of?

You probably will not have demonstrated all of the expectations listed, but have you hit the majority of the expectations?

How well? Did you develop the ability to meet these expectations independently? Did you challenge yourself?

## Curriculum Expectations I Believe I Have Met

### 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;

*My code uses several variables and @IBOutlets (below):*

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L13-L39>

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

*My code uses several assignment statements (below):*

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L25-L49>

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L69>

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L78>

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L92-L99>

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L105-L122>

**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;

*My code uses several comparison operators (below):*

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L69>

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L90-L96>

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L103>

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

*My code requires no arrays at this stage.*

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

*My code requires no arrays at this stage.*

### A2. Data Types and Expressions 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;

*The code uses sliders and buttons as input from the user.*

*The code determines a random value and uses ‘if’ statements to determine the meeting of certain expectations.*

*The code outputs results, as well as streak and score counts for the user to see.*

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

*The code uses sequence to perform actions at the desired time (this includes results being displayed in UI at time of input).*

*The code uses an @IBAction, which performs output only when called upon (below):*

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L66-L124>

*The code also uses ‘if’ statements inside of this @IBAction.*

**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).

*The code uses ‘if’ statements to determine higher and lower values (below):*

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L90-L113>

### 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;

*The code contains a random number generator (below):*

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L78>

**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.

*This code contains functions that are called upon by a sender (below):*

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L46>

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift#L116-L122>

### 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;

*My code had several syntax and logical errors, all were fixed by the IPO due date. A major error included a run-time error, in which my program (computer) crashed, and the application file was corrupted. This issue was fixed with the help of GitHub (and Mr. Gordon); an earlier (uncorrupt) version of my code was located and replaced my current version. Also, an issue occurred in which the program was unable to recognize variable changes until the next time called upon; this was fixed by re-ordering my code.*

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

*Links below will direct you to the commits in which major commenting was made and cleaning-up is most evident. Though I continued to clean my code throughout my commits (below):*

<https://github.com/rsgc-blackwell-s/algorithms-isp/commit/e4b2ff7573397ad7560b4f867da68289cf0fb778>

<https://github.com/rsgc-blackwell-s/algorithms-isp/commit/4b30bc7ebfab09efb71ddee50b45b068f99e845d>

**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);

*My code had several errors throughout, a common error was the ‘expected declaration’ message. You should notice by the lack of errors in my code now, that all errors were corrected.*

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

*I feel as though I consistently demonstrated my ability to trace errors in my code and correct run-time errors when they arrived. I also feel I discovered and learned correct logic and program flow while writing my code (a major example in section A4.1 of this document).*

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

*Although I never used Xcode’s version of app testing and staged development, I consistently used the built in ‘app simulator’ to test my application and locate issues within the code.*

### 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;

*With the help of the internet and my professor, I was able to solve problems at an excelled rate. When without my major resources, I used the common technique ‘trial and error’ most commonly and effectively. Also, ‘stepwise refinement’ and ‘working backwards’ were major keys in developing this application and fixing errors in my code.*

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

*Most of my work was independent, though I had the help of some of my peers as well. I outlined above (section B1.1), that I also had the help of Mr. Gordon and the internet.*

**B1.3** use the input-process-output model to solve problems.

*Most of this process was done with my own feedback, but I also received feedback from my professor.*

### B3. Designing Algorithms Use a variety of problem-solving strategies to solve different types of problems independently and as part of a team;

**B3.1** design simple algorithms according to specifications.

*The only way to exemplify this is with the following link (below):*

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/Odds!/ViewController.swift>

*Yes, that is the entire code. My whole code consisted of algorithms, all which follow specifications. Also, before beginning the development of my application I designed an algorithm to outline my desired output (below):*

<https://github.com/rsgc-blackwell-s/algorithms-isp/blob/master/ipo.txt>

## Final Comments and Proposal for Level of Achievement

Taking into consideration the purpose of this project and the evaluation criteria, what overall level of achievement do you feel you have earned?

I feel I took the extra step that many other students were afraid to take; by developing a working application in Xcode for iPhone users. I demonstrated all of the above expectations except for A1.5 and A1.6, involving the use of arrays; which my code has no purpose for at this stage of its development. I demonstrated my ability to work through any issue that arrived at a very determined rate. I also demonstrated my ability to ask for help when required. This project demanded that I acquire knowledge that was yet to be taught in the stage of our course. I learned more than any other student in the class; and I know that for a fact. I met all of my mini-goals (outlined on my sesame profile), and constantly committed my work to GitHub as asked. While you discover my application, notice every tedious detail, every adjustment made to the code, to the storyboard, to present a massive step to my skills in this ICS3U course, parallel to every requirement put in front of me.

I will continue to develop this app no matter how it is assessed, because it is the project I am proudest to have developed in all of my ICS years. I feel that I earned a level between a 4 and 4+, considering all aspects of this IPO.

Thank you for all your help,

Scotty Blackwell.