Rubric: Goal1: Assignment: The Duel - Part1

Programming for Web Applications 1

Percentage of Total Grade: 1%

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. You will submit your completed project via GIT.
2. You will need to ensure you have at least 6 reasonable commits.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
Player names	Two player names are accurately displayed and formatted in alert box	20
Round number	Round number is accurately displayed and formatted in alert box	20
Player health	Player health is accurately displayed and formatted in alert box	20
Random number	Correct random number used for damage	5
Code form	The two required functions (fight() and winnerCheck()) are employed effectively.	15
Executes no more than 10 rounds OR when there is a winner OR when both fighters die	The program should exit after a 10 rounds, OR when there is a winner, OR when both fighters die	20
DEDUCTIONS		
Functionality	5 points will be deducted for each occurrence of broken functionality or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list	-5
Instructions	5 points will be deducted for each occurrence where the instruction(s) were not followed.	-5
Comments	5 points will be deducted for code not properly commented	-5
Your course Professionalism grade is affected by your Investment grade.		

Rubric: Goal2: Assignment: The Duel - Part2

Programming for Web Applications 1

Percentage of Total Grade: 1%

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. You will submit your completed project via GIT.
2. You will need to ensure you have at least 3 reasonable commits.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
Player names	Two player names are accurately displayed and formatted in the alert box. Must use an array with index number to display the correct player's name.	10
Round number	Round number is accurately displayed and formatted in alert box	10
Player health	Must use an array with index number to display the correct Player's health in the alert box.	15
Random number	Must use an array with index number for the correct user's damage number which you will then use to generate the random number.	5
Array	An array for each fighter containing fighter's name, damage, and health is created. ex: var fighter1 = ["batman", 20, 100];)	25
Access Data	Using array access notation to access the fighter's data	25
Code form	The two required functions, fight() and winnerCheck() (from Duel1), are employed effectively.	10
DEDUCTIONS		
Functionality	5 points will be deducted for each occurrence of broken functionality or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list.	-5
Instructions	5 points will be deducted for each occurrence where the instruction(s) were not followed.	-5
Comments	5 points will be deducted for code not properly commented	-5
Your course Professionalism grade is affected by your Investment grade.		

Rubric: Goal2: Assignment: JavaScript Practice — Programming for Web Applications 1

Percentage of Total Grade: 1%

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. You will submit your completed project via GIT.
2. You will need to ensure you have at least 6 reasonable commits.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
create a function named "avgNumbers"	- accept 1 parameter into the function that will be an array of unlimited numbers - find the average of all the numbers - return the average from the function - console.log the answer outside of the function	15
create a function named "fullName"	- accept 2 parameters into the function that are strings (firstname and lastname) - return the name after it has been concatenated - console.log the answer outside of the function	15
create a function named "wordCount"	- accept 1 parameter into the function that is a long string of text words - create a function that counts all the words and return the answer - console.log the answer outside of the function	15
create a function named "charCount"	- accept 1 parameter into the function that is a long string of text - return length of the array of string characters - console.log the answer outside of the function	15
create a function named "vowelsInWord"	- accept 1 parameter into the function that is a a one word string - return the number of vowels in the word - console.log the answer outside of the function	15
create a function named "findNum"	- accepts 2 parameters into the function - 1. array of numbers, 2. boolean - if the second parameter being passed is "false" or null then ---- create an array with all of the odd numbers from the array - else ---- create an array with all of the even numbers from the array - return the array - console.log the answer outside of the function	25
DEDUCTIONS		
Functionality	5 points will be deducted for each occurrence of broken functionality or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list	-5

	Instructions	5 points will be deducted for each occurrence where the instruction(s) were not followed.	-5
	Comments	5 points will be deducted for code not properly commented	-5
	Your course Professionalism grade is affected by your Investment grade.		

Rubric: Goal3: Assignment: Debug

Programming for Web Applications 1

Percentage of Total Grade: 15%

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. You will submit your completed project into FSO.
2. Name your file lastname_firstname_debug.zip.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
DEDUCTIONS		
Syntax Errors (8 items)	Syntax errors occur before the JavaScript code event runs, basically meaning that the code can't compile. These errors (also known as parsing errors) occur when the programmer makes a typo JavaScript mistake (no closing string with quotes or escaping quotes with \, separating array values with a comma, missing necessary syntax characters as (), or {}.	-5 per item
Run-Time Errors (7 items)	The most common cause of runtime errors is when a variable or function does not exist (or the reference is misspelled). Script execution will stop. Incorrect Capitalization - not using Camel Case. Referencing code, functions or DOM objects before they exist. Missing parameter. Incorrectly typed commands.	-5 per item
Logical Errors (5 items)	Logic “errors” are the apparent lack of success (the desired effect does not happen). NO errors display. Simply programmers logic mistakes.	-5 per item
DEDUCTIONS		
Functionality	5 points are deducted for each occurrence of broken functionality or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list	-5
Instructions	5 points are deducted for each occurrence where the instruction(s) were not followed.	-5
Your course Professionalism grade is affected by your Investment grade.		

Rubric: Goal3: Assignment: Debug

Programming for Web Applications 1

Percentage of Total Grade: 15%

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. You will submit your completed project into FSO.
2. Name your file lastname_firstname_debug.zip.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
DEDUCTIONS		
Syntax Errors	Syntax errors occur before the JavaScript code event runs, basically meaning that the code can't compile. These errors (also known as parsing errors) occur when the programmer makes a typo JavaScript mistake (no closing string with quotes or escaping quotes with \, separating array values with a comma, missing necessary syntax characters as (), or {}.	-5 per item
Run-Time Errors	The most common cause of runtime errors is when a variable or function does not exist (or the reference is misspelled). Script execution will stop. Incorrect Capitalization - not using Camel Case. Referencing code, functions or DOM objects before they exist. Missing parameter. Incorrectly typed commands.	-5 per item
Logical Errors	Logic “errors” are the apparent lack of success (the desired effect does not happen). NO errors display. Simply programmers logic mistakes.	-5 per item
DEDUCTIONS		
Functionality	5 points are deducted for each occurrence of broken functionality or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list	-5
Instructions	5 points are deducted for each occurrence where the instruction(s) were not followed.	-5
Your course Professionalism grade is affected by your Investment grade.		

Rubric: Goal4: Assignment: The Duel - Part3

Programming for Web Applications 1

Percentage of Total Grade: 1%

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. You will submit your completed project via GIT.
2. You will need to ensure you have at least 6 reasonable commits.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
Object Creation	Using the fighter information from duel2, create an array of two objects with three keys per object (name, damage, health)	10
Fight Function	Modify the code from duel2 to reflect using the new objects. NO loop is to be used for this assignment. Since the button click will be the items that triggers the next round.	25
No Alerts	All alerts from duel2 must be removed from your file	5
Display Data in HTML	Use JavaScript's innerHTML property to change the text in the HTML. Display the information dynamically in the HTML (fighter's name and health at the top, current round number above the button)	25
Button Click	When the button is clicked (advance the round by one, the modified fight() function is called)	10
Game Over Items	Disable the button when the game is over. Display the appropriate "game over message" at the top. The message should be one of the following: "Fighter 1 wins", "Fighter 2 wins", "Both Fighters Die". Make sure the actual name of the fighter is shown not fighter1 or fighter2.	25
DEDUCTIONS		
Functionality	5 points are deducted for each occurrence of broken functionality or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list	-5
Instructions	5 points are deducted for each occurrence where the instruction(s) were not followed.	-5
Comments	5 points are deducted for code not properly commented	-5

	Your course Professionalism grade is affected by your Investment grade.				

Rubric: Goal4 Assignment: Guessing Game — Programming for Web Applications 1

Percentage of Total Grade: 1%

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. You will submit your completed project via GIT.
2. You will need to ensure you have at least 6 reasonable commits.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
Click event	Create an Event Listener on the guess button to listen for a click event. The guess button function, calls another function, to check if the data entered by the user is valid (this is graded below in Validate users input).	10
Random number	Create a random number between 1 & 10 is generated, and stored in a variable to be used in the "Correct guess" and the "Incorrect guess" items below.	10
Validate users input	A function is created to validate the users input: - validate that a number was entered. AND - validate that the number entered is between 1 and 10	20
Correct guess	Game ends correctly when you guess the computer's number (so the random number and the student's number matches): - display the appropriate successful message - deactivate the button by removing the event listener when the game is over (this is graded in "Remove event" below	20
Incorrect guess	A function is created to check if the student's answer is higher or lower and displays the appropriate message. The user then has to enter another guess (only allow 3 guesses)	15
Remove event	Once game is over, the button is disabled	10
Guess count	User is only allowed 3 attempts to guess correctly	15
EXTRA CREDIT: Keyboard event	A keyboard event for the ENTER button that also acts like clicking on the guess button. Keyboard Event functions correctly	10
DEDUCTIONS		
Functionality	5 points are deducted for each occurrence of broken functionality or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list	-5

	Instructions	5 points are deducted for each occurrence where the instruction(s) were not followed.		-5	
	Comments	5 points are deducted for code not properly commented		-5	
	Your course Professionalism grade is affected by your Investment grade.				

Rubric: Goal5 Assignment: Form Validation

Programming for Web Applications 1

Percentage of Total Grade: 1%

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. You will submit your completed project via GIT.
2. You will need to ensure you have at least 6 reasonable commits.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
userName input field	- The form field accepts a capitalized first character for the firstName and lastName. (i.e "John Doe", "Mary Ann Doe"). - An invalid input produces a red box. - An invalid input produces an error message. - A valid input produces a green box.	20
email input field	- The form field accepts a basic email address (i.e "jdoe@Fullsail.com"). - An invalid input produces a red box. - An invalid input produces an error message. - A valid input produces a green box.	15
phoneNumber input field	- The form field accepts a basic email address (i.e "(###)###-####"). - An invalid input produces a red box. - An invalid input produces an error message. - A valid input produces a green box.	10
socialSecurityNumber input field	- The form field accepts a social security number (i.e "###-##-####"). - An invalid input produces a red box. - An invalid input produces an error message. - A valid input produces a green box.	10
password input field	- The acceptable input to meet this requirement are as follows: the password's first character must be a letter, it must contain at least 4 characters and no more than 15 characters and no characters other than letters, numbers and the underscore may be used. - An invalid input produces a red box. - An invalid input produces an error message. - A valid input produces a green box.	30
onsubmit function	- Create an onsubmit function that contains a call to a function validateField. An argument is passed into the validateField function call. The argument will be the ID name of the input field. - Dynamically retrieve the ID name from the DOM/HTML - One call to the validateField function exists for each input field.	15

DEDUCTIONS		
Functionality	5 points are deducted for each occurrence of broken functionality or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list	-5
Instructions	5 points are deducted for each occurrence where the instruction(s) were not followed.	-5
Comments	5 points are deducted for code not properly commented	-5
Your course Professionalism grade is affected by your Investment grade.		

Rubric: Goal6 Assignment: Mid Term — Programming for Web Applications 1

Percentage of Total Grade: 30%

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. You will submit your completed project into FSO.
2. Name your file lastName_firstName_midterm.zip.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
Array of Objects	At least 2 array of objects are included in the main.js file, that reflects the example in the instructions.	10
Global Variables	No more than 4 Global Variables are included in the deliverable.	10
1st Console.log	Console.logs ALL the information in ALL objects on 3 lines.	5
Add to Array	A separate function is included that accepts key values (as the parameters) for a new object, and adds the new student information to the array of objects.	15
2nd Console.log	Console.logs ALL the information in ALL objects on 3 lines.	5
Button	Event Listener is correctly added to button and callback function executes correctly.	10
DOM .innerHTML	Displays ALL the information in ALL objects on 4 lines, in the HTML.	15
Average GPA	A separate function calculates the average GPA. This should not display the data.	15
Last Step	Disabled the onclick and change the text on the button.	5
Date Method	A date is added to the end of each student object, and the date displays in all the required outputs above.	10
DEDUCTIONS		
Functionality	5 points are deducted for each occurrence of broken functionality or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list.	-5

Instructions	5 points are deducted for each occurrence where the instruction(s) were not followed.	-5	
Don't Repeat Yourself (DRY)	10 points are deducted for each occurrence of duplicate code functionality in the main.js file.	-10	
Comments	5 points are deducted for code not properly commented	-5	
EXTRA CREDIT			
Validate GPA Format	Function validates a basic GPA format (#.##) without using Regular Expression.	10	
Your course Professionalism grade is affected by your Investment grade.			

Rubric: Goal7 Assignment: Objects

Programming for Web Applications 1

Percentage of Total Grade: 1%

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. You will submit your completed project via GIT.
2. You will need to ensure you have at least 6 reasonable commits.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
Array of Names	(main.js) Create an array called names that contains at least 5 people names.	2
People Array	(main.js) Create three instances of the Person object using a for loop and place each reference of the object in an array called people. - 3 person objects created - For loop is setup and configured correctly - The 3 person objects are stored in an array called people	15
Instantiate a Person Object	(main.js) When instantiating a Person object, make sure a randomly chosen name (using the Math.random() method) from the names array is sent to the constructor of the person along with what row number in the HTML the information will be displayed in. - randomly select a name from the names array using Math.random - pass the random name and row number to the constructor	10
populateHTML function	(main.js) Create a function called "populateHTML" which outputs the person's name and person's job, in the DOM.	5
no duplicate names	(main.js) Create code to NOT allow duplicate names to appear.	5
set up an Interval timer	(main.js) Set up an Interval that calls a runUpdate() function 30 times a second. Example: "setInterval(runUpdate, 1000 / 30);"	5
run prototype update to update the browser with new info	(main.js & person.js) Loop through each person and run the prototype update() function (in the person.js). This loop should be done in the runUpdate() function that is called from the setInterval (in the min.'s). For an example see instructions.	15
Array for "jobs" and "actions"	(person.js) Create 2 variables, "jobs" and "actions" which should be directly on the Person object (static variables). The jobs variable is an array of 4 or more jobs. The actions variable is an array of some actions a person could do.	3

Person Constructor	(person.js) A constructor named "Person" is created to create objects with the following keys: "name", "action", "job", "row" and to display the action results to the browser in the correct row. - "name": The name of the person. - "action": This property states what the person is actively doing and it is one of the values in the actions array. Randomly select one item from the Person.actions array for this property (use the Math.random() method). - "job": This property is set for the Person and is one of the values in the jobs array. You will randomly select one item from the Person.jobs array for this property(use the Math.random() method). - "row": The row number that is passed to the constructor. - Display the initial action of the person in 3rd column of the browser.	25
Update the "action"	(person.js) Called from the main.js file. The purpose of this prototype is to change the actions of the person every so often, this is based on the interval instructions for main.js. Needs to display the change of the "action" in the HTML in column 3.	15
DEDUCTIONS		
Functionality	5 points are deducted for each occurrence of broken functionality or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list.	-5
Instructions	5 points are deducted for each occurrence where the instruction(s) were not followed.	-5
Don't Repeat Yourself (DRY)	10 points are deducted for each occurrence of duplicate code functionality in the main.js file.	-10
Comments	5 points are deducted for code not properly commented	-5
EXTRA CREDIT		
Validate GPA Format	Function validates a basic GPA format (#.##) without using Regular Expression.	10
Your course Professionalism grade is affected by your Investment grade.		

Rubric: Goal8 Assignment: easyLibrary

Programming for Web Applications 1

Percentage of Total Grade: 1%

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. You will submit your completed project via GIT.
2. You will need to ensure you have at least 6 reasonable commits.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
Setup the main.js file	a. Start your file with a self executing function. b. Create a variable that stores ALL the anchor links in an array. (hint: use querySelectorAll) c. Create a variable that will store the returned value (or results) from a call to your library. Make a call to the ryu library and pass into it an argument. The argument should be the variable above that stores ALL the anchor links. d. Console.log the results that are sent back from the library using the variable from "c." (directly above) e. Make a call to your ryu library and pass into it the variable that stores all the anchor links (the variable from "b.", above), and runs the .each prototype function that is in your library.	35
Setup the RYU library file	a. Your ryu.js file includes a template of a library. There are two sections, the Constructor and the Prototype. b. Setup the Constructor and include a parameter that will be used throughout the library. The main.js file will be passing an argument to this library. c. Setup the Prototype section. Make the necessary changes to the template to ensure the Constructor can use the Prototype. d. Create within the Prototype section a function named "init", and another called "each" e. The "init" function will accept a parameter from the Constructor. Set the Prototypes ".elements" property to equal the parameter being passed into the "init" function. f. Setup the "each" function with a "FOR" loop that when runs, console.log's the Prototype's "this.element" property. The property item within the array needs to console.log based on the index number of the "FOR" loop.	35
Change the background color of the links	Change the background color of your links to Gray	10
Create an onclick event	Add an "onclick" event to the ryu .each function. When the "onclick" occurs, console.log the "THIS" item.	20
DEDUCTIONS		

Functionality	5 points are deducted for each occurrence of broken functionality or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list.	-5	
Instructions	5 points are deducted for each occurrence where the instruction(s) were not followed.	-5	
Comments	5 points are deducted for code not properly commented	-5	
Your course Professionalism grade is affected by your Investment grade.			

Rubric: Goal9 Assignment: Canvas

Programming for Web Applications 1

Percentage of Total Grade: 1%

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. You will submit your completed project via GIT.
2. You will need to ensure you have at least 6 reasonable commits.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
Dynamic Canvas	Create the Dynamic Canvas using the information in the instructions	5
Data Array	Data for chart should be in an array	10
Background Image	Background Image must be behind data	10
Bar Chart	Minimum of 4 bars	10
Bar Chart Scales	Bars must scale correctly using the data in the "Data Array"	25
Chart Name	Chart should be named accordingly	10
Top of Bar Label	The top of each bar should be labeled and positioned correctly using the numbers being diagrammed from the "Data Array"	10
Bottom of Bar Label	The bottom of each bar should be labeled and positioned correctly using the numbers in the "xAxisData" array	10
Functions	The "drawBars" and "drawChartText" functions are created using the instructions	10
DEDUCTIONS		
Functionality	5 points are deducted for each occurrence of broken functionality or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list.	-5
Instructions	5 points are deducted for each occurrence where the instruction(s) were not followed.	-5

	Comments	5 points are deducted for code not properly commented		-5	
	Your course Professionalism grade is affected by your Investment grade.				

Rubric: Wk4 Assignment FINAL

Programming for Web Applications 1

Percentage of Total Grade: 35%

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. You will submit your completed project into FSO.
2. Name your file lastName_firstName_FINAL.zip.

Item	*DESCRIPTION*	*Points*
PWA1: Requirements		
Create Objects w/ Constructor	The constructor will create each object, and the object will programmatically be stored in the students array.	25
1st Console.log	Console.logs ALL the information in ALL objects on 4 lines (Name, Address, GPA, Date)	5
Create Another Object w/ Constructor	The constructor will create the object, and the object will programmatically be stored in the students array.	15
2nd Console.log	Console.logs ALL the information in ALL objects on 4 lines (Name, Address, GPA, Date)	5
Average GPA w/ Prototype	A prototype method is used to calculate each students average GPA.	25
All other items from the Mid Term are Functioning	All other requirements from the Mid Term are in place and working as expected	25
DEDUCTIONS		
Functionality	5 points are deducted for each occurrence of broken functionality, missing items, or errors that may or may not be covered in this rubric. Also a deduction is issued for improper uploading and omissions from the criteria list.	-5
Instructions	5 points are deducted for each occurrence where the instruction(s) were not followed	-5
Don't Repeat Yourself (DRY)	10 points are deducted for each occurrence of duplicate code functionality in all JavaScript files.	-10
Comments	5 points are deducted for code not properly commented	-5
EXTRA CREDIT		

	Validate GPA Format	Function validates a basic GPA format (#.##) USING Regular Expression.		10	
	Your course Professionalism grade is affected by your Investment grade.				