

# CSCE 120: Learning To Code

## Processing Data I Hacktivity 8.2

### Introduction

Prior to engaging in this hacktivity, you should have completed all of the pre-class activities as outlined in this module. At the start of class, you will be randomly assigned a partner to work with on the entirety of this hacktivity as a *peer programming* activity. Your instructor will inform you of your partner for this class.

One of you will be the driver and the other will take on the navigator role. Recall that a driver is in charge of the keyboard and computer while the navigator is in charge of the handout and directing the activity. However, you are *both* responsible for contributing and discussing solutions. If you were a driver/navigator in the prior activity, switch roles for this activity.

You will use the same project from the previous Hacktivity. If you need to, you can re-download it from GitHub using the URL, <https://github.com/cbourne/LoopProject>.

### 1 Processing Records

We have included the the enrollment data used in a previous activity and in the demonstrations (see `enrollment/roster.js`). Write a loop that validates the data by finding any *duplicate* records. An enrollment record is a duplicate if it has the same student (identified with an NUID) enrolled in the same class (Course Number).

Write code to find and print (to the console) any duplicate records.

## 2 Using Loops to Manipulate the DOM

In these exercises, you'll write loops that not only process data, but manipulate the DOM at the same time.

1. Open the files in the `multiplicationTable` folder. The HTML page has an input box where the user can enter an integer  $n$  and click the "Generate" button. When clicked, it calls the `generate()` function. Complete the function by writing code to generate an  $n \times n$  multiplication table. For example, for  $n = 5$  it should look something like this:

### Multiplication Table

Enter an integer to generate a multiplication table below.

	1	2	3	4	5
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25

2. Open the files in the `fruitForm` folder. In the JavaScript file there is an array of strings. Write code (in a `ready` function so that it executes when loaded) that process the array of strings and dynamically creates a radio button group out of their values. Use the name `fruit` and the values for each button should correspond to the index of the element in the array. When completed, it should look something like this:

### Favorite Fruit:

- Pineapple
- Pear
- Apple
- Banana
- Grapes

You can use the "Test" button to see if your code is implemented properly.