

## Overview

Welcome to CSE-021! This lab is meant to be hard and to test your knowledge from CSE-020. Perhaps it has been awhile since you took it or what you retained is shaky. This lab should not take the full three hours but if it does then you need to start studying. CSE-021 is going to be decidedly harder than CSE-020 and we expect you to have certain level of competence with the material covered already. Take this lab as the barometer of the comfort level we are expecting incoming students for this class. We encourage the students to work together so feel free to do that in lab sessions or outside, but you are not allowed to share code. You should list the people you helped, helped you or just worked together.

## Getting Started

After starting Eclipse, create a new project called Lab21\_01. Import `FindDuplicateCount.java` from the assignment page (right click the file and save into the project directory).

### Part 1: Create RemainderFunc.java

This program asks for two inputs from the user:

- Max number, `maxnum`
- Divisor, `divisor`

Then it displays all numbers that are multiples of `divisor` starting at 1 up to `maxnum` (inclusive) as shown in the sample runs below. You must use the remainder (%) operation for this task.

**Sample Runs (user input shown in blue, with each run separated by a dashed-line):**

```
Please enter the max number: 10
Please enter the divisor: 2
Multiples of 2 between 1 and 10 (inclusive) are:
2
4
6
8
10
-----
Please enter the max number: -2
Invalid input. Please enter a valid max number (>= 0): -8
Invalid input. Please enter a valid max number (>= 0): 10
Please enter the divisor: 3
Multiples of 3 between 1 and 10 (inclusive) are:
3
6
9
-----
Please enter the max number: 8
Please enter the divisor: -2
Invalid input. Please enter a valid divisor (> 0): 0
Invalid input. Please enter a valid divisor (> 0): -5
Invalid input. Please enter a valid divisor (> 0): 10
Multiples of 10 between 1 and 8 (inclusive) are:
No number were found.
-----
Please enter the max number: -9
Invalid input. Please enter a valid max number (>= 0): -10
Invalid input. Please enter a valid max number (>= 0): 25
Please enter the divisor: 0
```

```
Invalid input. Please enter a valid divisor (> 0): 0
Invalid input. Please enter a valid divisor (> 0): -8
Invalid input. Please enter a valid divisor (> 0): 4
Multiples of 4 between 1 and 25 (inclusive) are:
4
8
12
16
20
24
```

## Part 2: Fill-in FindDuplicateCount.java

You are given an integer array (`arr`) declared in the program. Count all the numbers that are repeated in the array entries. If there are no duplicates then your program should output should print out according to the expected output below. If there is only one duplicate then the message should state that. Finally, for two or more duplicates then the output should match the expected output. Your solution should use loops (hint: nested loops) and should work for any integer array.

### Expected Output:

```
There are no duplicates with value 1 beyond index 0
There are 3 more occurrences of value 2 starting at index 1
There are 2 more occurrences of value 2 starting at index 2
There are 2 more occurrences of value 3 starting at index 3
There is only 1 more occurrence of value 4 starting at index 4
There is only 1 more occurrence of value 2 starting at index 5
There are no duplicates with value 4 beyond index 6
There is only 1 more occurrence of value 3 starting at index 7
There are no duplicates with value 0 beyond index 8
There are no duplicates with value 5 beyond index 9
There are no duplicates with value 3 beyond index 10
There are no duplicates with value 2 beyond index 11
```

## Part 3: (Assessment) Logic Check

Create a Word document or text file named **Part3** that contains answers to the following:

1. When did you take CSE-020?
2. How long did this lab take to finish?
3. Any difficult topics for you in these two exercises? If so list them:
4. For the array given in **FindDuplicateCount.java**, which index is the last check we need to make to count all the duplicates?
5. Interpret the number 101010 in bases 2, 10, 16. For each, convert to the other two bases (for example, when interpreting as base 2, convert to bases 10 and 16; when interpreting as base 10, convert to bases 2 and 16; etc.).

## What to hand in

When you are done with this lab assignment, submit all your work through CatCourses.

**Before** you submit, make sure you have done the following:

- Attached the file named **Part3** containing answers to Assessment questions (1 – 5).
- Attached created **RemainderFunc.java** and filled in **FindDuplicateCount.java** files.
- Filled in your collaborator's name (if any) in the "Comments..." text-box at the submission page.

Also, remember to demonstrate your code to the TA or instructor before the end of the grace period.