

CS103 – Fall 2025

Lab05 Exercises

Objectives:

- To practice using loops, conditionals, and functions in Python.
- To reinforce list manipulation and random number generation.
- To solve real-world style problems with arithmetic and logical reasoning.

General Instructions

1. In today's lab, you are required to complete all required exercises during class to receive attendance credit.
2. You are welcome to:
 - Work together with classmates.
 - Search online for help or documentation.
 - Ask the TA for guidance if you are stuck.
3. To receive credit, you must finish and show your solutions to the TA before leaving lab.
4. There are also extra (optional) exercises at the end for those who want to challenge themselves. These are not required for attendance credit but are recommended for practice.

Exercise Instructions

- Make a folder **Lab5** inside your **cs103fa25** folder.
- Create a new notebook inside your Lab5 folder (**lab05.ipynb**).

Required Exercises

EXERCISE 1:

Write the function “**evenNumbers**” that takes an int “**x**” and prints all even numbers between 0 and **x**.

Sample Input

x = 12

Expected Output

0 2 4 6 8 10 12

EXERCISE 2:

Write a function “**factorial**” that receives an int “**f**” and returns **f!** (f factorial) (Do not use `math.factorial` function).

Sample Input

f = 6

Expected Output

720

EXERCISE 3:

Write a function “**randNumbers**” that receives an int “**r**” and generates **r** different random numbers. The randomized number must be $0 \leq r < 10$

The function prints each generated random value and their average. Consider that randomized numbers may be different in every call because of that with the same **r** value, you may get different values and different averages than the examples below.

1. Sample Run

Example program run for **randNumbers(5)**

1. Random Number: 4
2. Random Number: 4
3. Random Number: 3

4. Random Number: 7

5. Random Number: 2

The average of these random numbers is 4

2. Sample Run

Example program run for randNumbers (3)

1. Random Number: 2

2. Random Number: 0

3. Random Number: 1

The average of these random numbers is 1

Extra (Optional) Challenges

These are not required for credit but will help you practice and strengthen your problem-solving skills.

Challenge 1: evenOddBalance(n)

Write a function evenOddBalance(n) that:

- Takes an integer n and examines all numbers from 1 to n.
- Sums up all even numbers and all odd numbers separately.
- Returns a string describing which sum is greater or if they are equal.

Sample Function Call:

```
>>> evenOddBalance(10)  
"Even sum is greater: 30 vs 25"  
>>> evenOddBalance(5)  
"Odd sum is greater: 9 vs 6"
```

Challenge 2: listDifference(list1, list2)

Write a function **listDifference(list1, list2)** that returns a **new list** containing all numbers that are present in list1 but **not** in list2.

- Do **not** use Python's built-in set operations. Use loops and if/else.

Sample Function Call:

```
>>> listDifference([1,2,3,4], [2,4,6])  
[1, 3]  
>>> listDifference([10,20,30], [5,10,15])  
[20, 30]
```

Challenge 3: mountainPeak(nums)

Write a function **mountainPeak(nums)** that returns **True** if the list forms a "mountain":

- Strictly increasing up to a single peak,
- Then strictly decreasing after the peak.

Return **False** otherwise.

(Hint: You will need loops and careful comparisons.)

Sample Function Calls:

```
>>> mountainPeak([1, 3, 5, 4, 2])
```

```
True
```

```
>>> mountainPeak([2, 2, 3, 4])
```

```
False
```

```
>>> mountainPeak([1, 2, 3])
```

```
False
```

Challenge 4: Find All Disappeared Numbers

You are given a list of integers **nums** of length **n**, where each integer is between 1 and **n** (inclusive).

Some numbers appear **once**, some appear **twice**, and some numbers from 1 to **n** may be **missing**.

Task:

Write a function **findDisappearedNumbers(nums)** that returns a **list of all numbers between 1 and n that do not appear in nums**.

- Do **not** use Python sets or dictionaries.
- Use only **loops, lists, and if/else** to solve.
-

Sample Function Calls:

```
Input:  nums = [4,3,2,7,8,2,3,1]
```

```
Output: [5,6]
```

```
# Explanation: Numbers 5 and 6 are missing.
```

Input: nums = [1, 1]

Output: [2]

Constraints

- $1 \leq \text{len}(\text{nums}) \leq 10^5$
 - $1 \leq \text{nums}[i] \leq \text{len}(\text{nums})$
-

To get attendance credit, finish Exercises 1–3.

If you finish early, try the optional challenges!