EEC 351 Python Programming Homework

Suggested Completion: **Before Week 3 Begins**

1. Overview

This self-paced assignment is designed to refresh your `Python` skills before we dive into machine learning[1] . These exercises help you practice syntax, libraries, and best practices relevant to the course. You are encouraged to attempt them for your own confidence and readinessno submission is required.

Learning Objectives

By the end of this self-practice, you should be able to:

1. Write clean, well-documented `Python` scripts.
2. Manipulate arrays with `NumPy` and datasets with `Pandas`.
3. Visualize data using `Matplotlib`.
4. Implement simple neural-network layers with `PyTorch`.

Prerequisite Checklist

- Basic programming concepts (variables, loops, functions).
- Familiarity with command line or an IDE (VS Code / PyCharm)– A good chance to learn if not familiar yet.
- Git basics (clone, commit, push)– Again, a good chance to learn if not familiar yet.

2. Resources

2.1 Official Documentation

- NumPy
- Pandas
- Matplotlib
- PyTorch

2.2 Curated Exercise Sets

- 101 NumPy Exercises
- 80 Pandas Exercises
- Matplotlib Beginner Tutorial
- PyTorch Tutorial Notebook

[1] We have used Perplexity Labs for resource lookup and idea validation; content was manually written.

3. Self-Practice Tasks

3.1 Part A Core Exercises

Try the following exercises at your own pace. These foundational problems are designed to sharpen your basic Python skills including conditionals, functions, comprehensions, file handling, and testing.

Task

A1. FizzBuzz++

The classic "FizzBuzz" task is often used in programming interviews. Traditionally, you print numbers from 1 to `n`, but replace:

- Multiples of 3 with `"Fizz"`,
- Multiples of 5 with `"Buzz"`,
- Multiples of both 3 and 5 with `"FizzBuzz"`.

Task: Write a function `fizzbuzz_plus(n, k=7)` that extends this idea by printing `"Boom"` for multiples of `k`, in addition to the standard FizzBuzz rules.

Example Output for `n=21, k=7`:

```
1
2
Fizz
4
Buzz
Fizz
Boom
8
Fizz
Buzz
11
Fizz
13
Boom
FizzBuzz
16
17
Fizz
19
Buzz
FizzBoom
```

Also, write unit tests using `pytest` to verify correctness.

Task

A2. List Comprehension Drills

List comprehensions offer a concise way to create lists. This exercise tests your ability to combine filtering and transformation in a single line of Python.

Task: Write a function that:

- Takes a list of integers (e.g., `[4, 5, 6, 7, 11, 12]`)
- Filters the prime numbers
- Returns their squares in a list

You may define a helper function `is_prime(n)` and then apply a list comprehension to filter and transform.

Expected Output Example:

```
Input: [4, 5, 6, 7, 11, 12]
Output: [25, 49, 121]
```

Task

A3. File I/O and Exceptions

This task helps you practice reading files, parsing data, using Pythons built-in libraries like `csv` and `statistics`, and writing robust code with exception handling.

Task:

- Read a file `data/grades.csv` that contains rows like: `Alice,89,92,85`
- Compute the mean, median, and standard deviation of scores for each student
- Output the results to a JSON file `summary.json` with the format:

```
{
  "Alice": {"mean": 88.67, "median": 89.0, "std": 3.51},
  ...
}
```

- Skip and log any malformed lines (e.g., non-numeric grades or missing values)

Use the `statistics` and `json` modules. Aim to write clean, modular code.

3.2 Part B Data Manipulation with NumPy & Pandas

Task

B1. NumPy 101

Solve exercises 130 from the 101 NumPy Exercises list. Use a Jupyter notebook to try and track them.

Task

B2. Pandas Challenge

Using `pandas`, clean the `titanic.csv` dataset, then compute and visualize:

a) Survival rate by passenger class.

b) Correlation heatmap of numeric features.

Reflect on any dataquality issues you encountered.
Dataset Links:

- Download the Titanic CSV directly: Dataset
- Alternatively, grab it from Kaggle (includes train/test split): Here

3.3 Part C Visualization with Matplotlib

Task

C1. Reproduce Figure

Recreate Figure from task B2 from the Matplotlib tutorial using the provided synthetic data in `data/figure_data.npz`. Match colors, labels, and styles.

Suggested Data Source: You can use the NumPy '.npz' file with paired 'x', 'y' arrayse.g., generated via:

```
import numpy as np
np.savez('data/figure_data.npz', x=x_values, y=y_values)
```

3.4 Part D Mini-Project

Neural-Net Sandbox: Implement a 2-layer feed-forward network in PyTorch to classify Fashion-MNIST. Aim for more than 85% accuracy.
There are many examples of such projects available, it is a good idea to do a few of them.

4. How to Practice Efficiently

- Create a local folder named `python-practice/`.
- Organize your work using subfolders like `partA/`, `partB/`, etc.
- Use `git` locally or a GitHub repo if you wish to track your progress.
- Follow good coding practicesadd comments, modularize, and write test cases.
- Share your code on the discussion forum or offline to peers if you'd like feedback.

5. Honor Code

While this is a Homework and submission is not required, we encourage you to try the tasks independently to get the most benefit. Use external resources for learning, but make sure you understand each solution you write.