Python/Java Problem Set Worksheet 1.H CMPE 261 Large Scale Programming

Osman Emre Tutay

December 2025

Problem 1: Python File Handling

Write a Python program to manage student grades using file I/O operations:

- Create a text file named `grades.txt` and write three lines of data (format: `Name,Grade`).
- Read the file, calculate the average grade, and append the average to the end of the file.

Instructions:

- Use the `open()` function with appropriate modes ('w', 'r', 'a').
- Ensure the final file contains the original names and the calculated average.

Sample Output (Console):

```
File created and processed.
Average grade: 85.0 appended to file.
```

File Content (grades.txt after execution):

```
Alice,80
Bob,90
Charlie,85
Average: 85.0
```

Problem 2: Python Exception Handling (List Access)

Write a Python program that safely retrieves elements from a predefined list.

- Create a list named `fruits` containing 5 strings (e.g., "Apple", "Banana", "Orange", "Grape", "Mango").
- Ask the user to input an index number to select a fruit.
- Print the fruit at the selected index.
- Use `try-except` blocks to handle:
  - `IndexError`: If the user enters an index outside the range of the list.
  - `ValueError`: If the user enters a non-integer value (e.g., text).

Sample Input 1:

```
Enter index: 10
```

Sample Output 1:

```
Error: Index out of bounds. Please choose between 0 and 4.
```

Sample Input 2:

```
Enter index: 2
```

Sample Output 2:

```
You selected: Orange
```

Problem 3: Java Conditional Statements and Loops

Write a Java program to analyze an array of integers:

- Iterate through the array using a `for` loop.
- Use conditional statements (`if-else`) to classify each number as "Positive", "Negative", or "Zero".
- Print the count of each category at the end.

Instructions:

- Define the array explicitly in the `main` method (e.g., {`-5, 0, 10, 3`}).

Sample Output:

```
Processing: -5 -> Negative
Processing: 0 -> Zero
Processing: 10 -> Positive
Processing: 3 -> Positive
Total Positive: 2
Total Negative: 1
Total Zero: 1
```

Problem 4: Java Classes with Constructors

Design a Java class named `Car` representing a vehicle.

- **Attributes:** `make` (String), `model` (String), and `year` (int).
- **Constructor:** A parameterized constructor to initialize all three attributes.
- **Method:** `displayInfo()` to print the car's details in a formatted string.

Instructions:

- In the `main` method, create two distinct `Car` objects and call `displayInfo()` for both.

Sample Output:

```
Car 1: Toyota Corolla (2020)
Car 2: Ford Mustang (2022)
```

Problem 5: Multiple Classes

Create a Java system with two classes: `Author` and `Book`.

- **Author Class:** Contains a `name` (String) and a constructor to set it.
- **Book Class:** Contains `title` (String) and an object of type `Author`.
- The `Book` constructor should accept a title and an `Author` object.
- Create a method in `Book` to print: `"Book: [Title] by [Author Name]"`.

Instructions:

- Instantiate an `Author` first, then pass it to a new `Book` object in the `main` method.

Sample Output:

```
Book: Harry Potter by J.K. Rowling
```

Problem 6: Java Static Methods

Create a utility class named `MathHelper` that does not need to be instantiated.

- Define a **static** method `calculateCube(int n)` that returns the cube of the number (n^3).
- Define a **static** method `isEven(int n)` that returns `true` if the number is even, otherwise `false`.

Instructions:

- Call these methods directly from the `main` method without creating an object of `MathHelper`.

Sample Input:

```
Number: 4
```

Sample Output:

```
Cube of 4 is: 64
Is 4 even? true
```