FINAL PYTHON PROGAMMING EXAM

Scenario 1: File Processing and Exception Handling

You are tasked with creating a program to manage student records. The program should read data from a text file containing student names and their corresponding grades (each student's name and grade separated by a comma). After reading the data, the program should calculate the average grade and display it to the user. However, the program should handle potential errors gracefully.

Tasks:

1. Create a Python program that reads student records from a text file.
2. Implement exception handling to catch file not found errors and handle them appropriately.
3. Calculate the average grade of the students and display it to the user.
4. If any student's grade is not a valid numerical value, raise a custom exception and handle it by displaying an error message.

student_records.txt

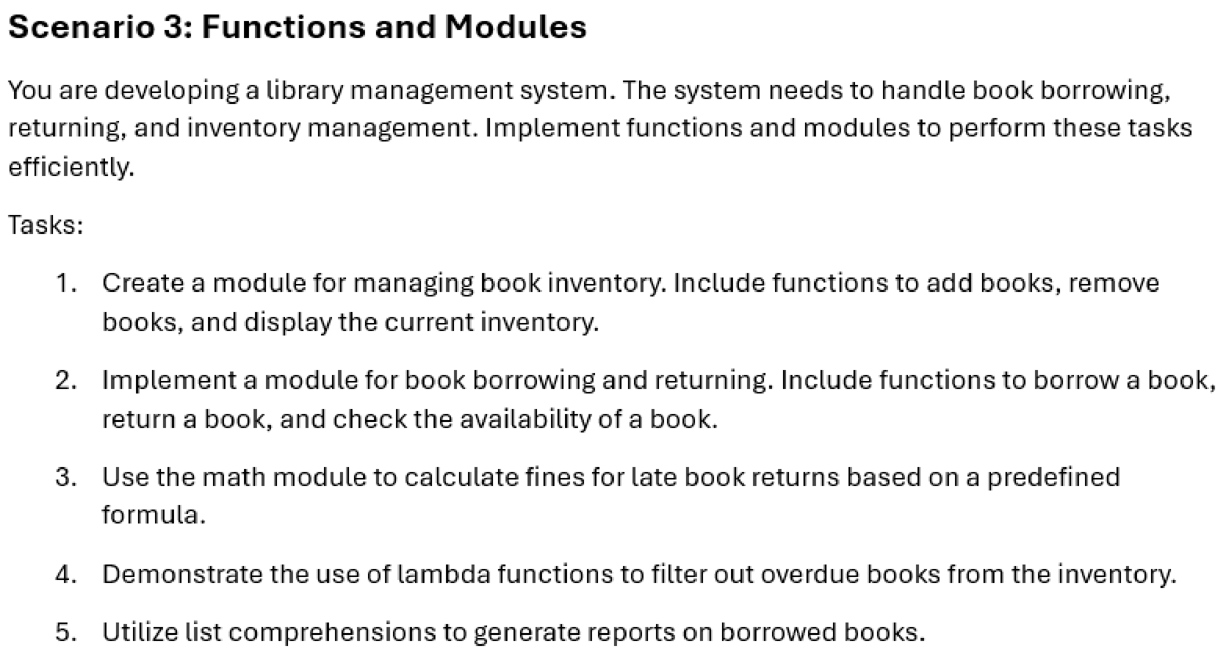
```
Alice, 85
Bob, 90
Charlie, 75
Dave, 82
Eve, 95
Frank, 78
Grace, Invalid Grade
```

Scenario 2: Object-Oriented Programming and Inheritance

You are building a system to manage different types of vehicles in a transportation company. There are three types of vehicles: Car, Truck, and Motorcycle. Each vehicle has common attributes like make, model, and year, as well as specific attributes like number of doors for a car, cargo capacity for a truck, and type of drive for a motorcycle. Implement classes for each type of vehicle with appropriate attributes and methods.

Tasks:

1. Create a base class called Vehicle with common attributes and methods.
2. Implement classes for Car, Truck, and Motorcycle, inheriting from the Vehicle class.
3. Include methods in each subclass to display vehicle information and perform any specific actions (e.g., start engine, accelerate).
4. Demonstrate the use of inheritance by accessing both the common and specific attributes/methods of each vehicle type.

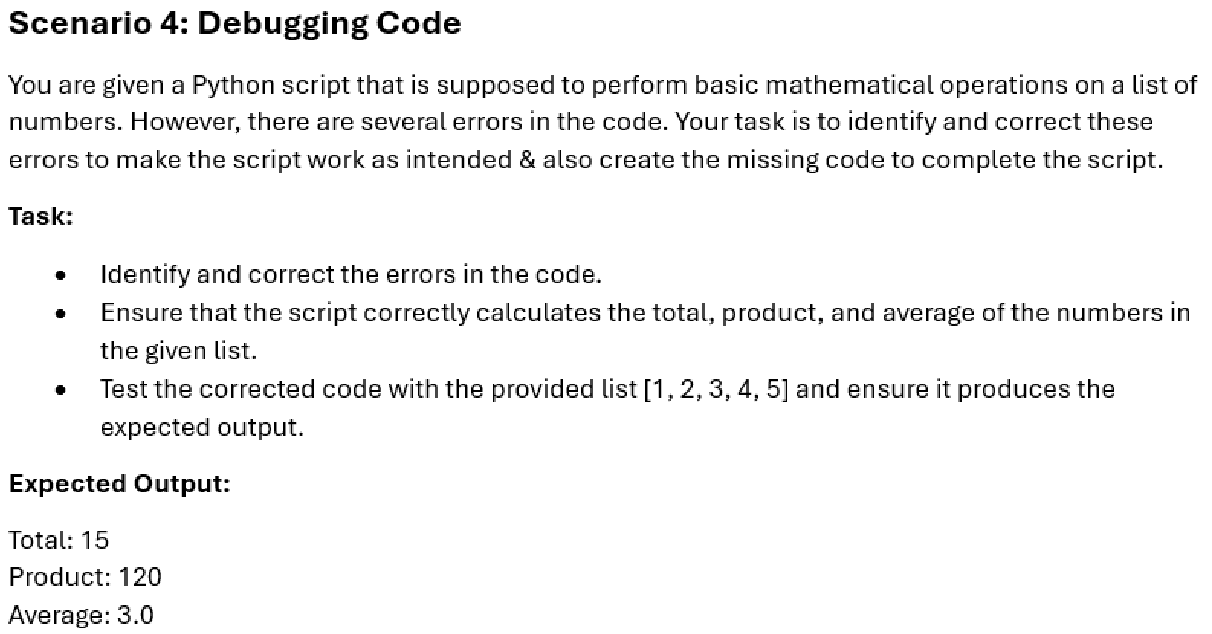


Scenario 3: Functions and Modules

You are developing a library management system. The system needs to handle book borrowing, returning, and inventory management. Implement functions and modules to perform these tasks efficiently.

Tasks:

1. Create a module for managing book inventory. Include functions to add books, remove books, and display the current inventory.
2. Implement a module for book borrowing and returning. Include functions to borrow a book, return a book, and check the availability of a book.
3. Use the math module to calculate fines for late book returns based on a predefined formula.
4. Demonstrate the use of lambda functions to filter out overdue books from the inventory.
5. Utilize list comprehensions to generate reports on borrowed books.



Scenario 4: Debugging Code

You are given a Python script that is supposed to perform basic mathematical operations on a list of numbers. However, there are several errors in the code. Your task is to identify and correct these errors to make the script work as intended & also create the missing code to complete the script.

Task:

- Identify and correct the errors in the code.
- Ensure that the script correctly calculates the total, product, and average of the numbers in the given list.
- Test the corrected code with the provided list [1, 2, 3, 4, 5] and ensure it produces the expected output.

Expected Output:

```
Total: 15
Product: 120
Average: 3.0
```

Faulty Python Code (debugging_script.py):

```
def perform_operations(numbers)

  total = 0

  product = 1

  for num in numbers:
      total *= num
      product += num
  average = total / len(numbers)
  return total, product, average
```