Data Structure

Problem:

Given a nested dictionary that represents a student's course enrollment and grades. Write a Python function called `calculate_gpa` that takes the `student_data` dictionary as input and calculates the Grade Point Average (GPA) for the student. The GPA is calculated as the sum of the product of each course grade and its corresponding credits, divided by the total number of credits.

```python
student_data = {
    "student_id": 123456,
    "name": "John Doe",
    "courses": {
        "Math": {
            "grade": 3.8,
            "credits": 3
        },
        "Science": {
            "grade": 3.5,
            "credits": 4
        },
        "English": {
            "grade": 3.0,
            "credits": 3
        },
        "History": {
            "grade": 3.3,
            "credits": 3
        }
    }
}

gpa = calculate_gpa(student_data)
print("The GPA is:", gpa)
```

The GPA is: 3.41

Interpreting Exception

Problem:

(a) Read the traceback carefully and identify the type of exception that occurred, the line number and function name where the exception was raised, and the specific cause of the exception.

```python
def calculate_average(numbers):
    total_sum = sum(numbers)
    average = total_sum / len(numbers)
    return average

numbers = [5, 10, 15, 20, "25"]
result = calculate_average(numbers)
print("The average is:", result)
```

```
Traceback (most recent call last):
  File "example.py", line X, in <module>
    result = calculate_average(numbers)
  File "example.py", line Y, in calculate_average
    total_sum = sum(numbers)
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

(b) How would you fix this error?

Debugging

Problem:

(a) The following Python code is intended to find the maximum value in a list of numbers. However, there is a bug in the code that prevents it from producing the correct result. Your task is to identify and fix the bug.

```python
def find_maximum(numbers):
    max_value = 0
    for number in numbers:
        if number > max_value:
            max_value = number
    return max_value

numbers = [4, 9, 3, 6, 12, 7, -5]
result = find_maximum(numbers)
print("The maximum value is:", result)
```

(b) What test cases could you write to make sure your fixes are correct.

Testing

Problem:

Given an implementation of a Python function called `calculate_factorial` that calculates the factorial of a given non-negative integer, create at least three test cases to verify its accuracy. Your test cases should cover a range of input values and include edge cases.

```python
def calculate_factorial(n):
    if n < 0:
        return None
    if n == 0:
        return 1
    else:
        result = 1
        for i in range(1, n + 1):
            result *= i
        return result

# Example usage
number = 5
factorial = calculate_factorial(number)
print(factorial)
```

Graph

Problem:

You are given a graph representing a social network. Each node in the graph represents a person, and each edge represents a friendship connection between two individuals. Your task is to find the person with the highest number of friends in the social network.

```python
import networkx as nx

# Create an empty undirected graph
social_network = nx.Graph()

# Add nodes representing individuals
social_network.add_nodes_from(["Alice", "Bob", "Charlie", "David", "Emma"])

# Add edges representing friendships
# add_edges_from() adds all the edges from the provided list into the graph that the function is being called on.
social_network.add_edges_from([("Alice", "Bob"), ("Bob", "Charlie"), ("Charlie", "David"), ("David", "Emma"), ("Emma", "Alice")])
```

Class

Problem:

You are given a Python class representing a bookstore inventory. Examine the given classes and write down the output that will be generated when the code snippet is executed.

```python
class Book:
    def __init__(self, title, author, price):
        self.title = title
        self.author = author
        self.price = price

    def display_info(self):
        print("Title:", self.title)
        print("Author:", self.author)
        print("Price:", self.price)

class Bookstore:
    def __init__(self):
        self.inventory = []

    def add_book(self, book):
        self.inventory.append(book)

    def display_inventory(self):
        print("Bookstore Inventory:")
        for book in self.inventory:
            book.display_info()

bookstore = Bookstore()

book1 = Book("The Great Gatsby", "F. Scott Fitzgerald", 10.99)
bookstore.add_book(book1)

book2 = Book("To Kill a Mockingbird", "Harper Lee", 7.99)
bookstore.add_book(book2)

book3 = Book("Pride and Prejudice", "Jane Austen", 6.99)
bookstore.add_book(book3)

bookstore.display_inventory()
```

Function

Problem:

Give two reasons for using functions

Sharing (by reference)

Problem:

Examine the given code and write down the output that will be displayed when the code snippet is executed.

```python
def modify_data(data):
    data["list"].append(4)
    data["nested_dict"]["key"] = "new value"
    data = {"list": [1, 2, 3], "nested_dict": {"key": "value"}}

data_structure = {"list": [5, 6, 7], "nested_dict": {"key": "old value"}}
modify_data(data_structure)
print(data_structure)
```

Performance

Analyze the runtime performance of the following two implementations of the same function. Identify the implementation that is more efficient and explain why it is more performant?

```python
def first_implementation(nums, target):
    for i in range(len(nums)):
        for j in range(i + 1, len(nums)):
            if nums[i] + nums[j] == target:
                return [i, j]
    return []

def second_implementation(nums, target):
    num_map = {}
    for i, num in enumerate(nums):
        complement = target - num
        if complement in num_map:
            return [num_map[complement], i]
        num_map[num] = i
    return []
```

Everything is generated using ChatGPT with the following procedure and format:

1. Providing the syllabus as context to Chatgpt
2. Provide as lecture slide as possible in form of text
3. Ask the question:
    a. Given the context of CSE160, write an intro to python programming problems involving [topic] that is solvable by hand without running the code. I want students to [more specification]
    b. *To change the content of a problem. Make this problem slightly harder by making the [topic] more complicated, and I don't want students to [more specification]
4. Ask solution and explanation:
    a. For this problem, give a solution and an explanation.