National University of Computer and Emerging Sciences
Karachi – Campus

Introduction to Information and Communication Technologies (CL1000)

Lab Final-Exam

Total Time (Hrs): 02
Total Marks: 50
Total Questions: 04

Paper A

Date: December 05, 2024
Time: 08:30 AM – 10:30 AM

Course Instructor(s)
Mr. Ghulam Qadir Bhurgari, Miss. Khadija tul Kubra

____________________	______________	____________________
Student Name	Section	Roll Number

Q1: [illegible: 16]

You are tasked with creating an **Inventory Management System** that allows users to manage products in a store, update product quantities, and calculate the total value of products in stock. The system tracks the products, their quantities, and their prices, and allows users to interact with them using simple functions.

Requirements:

1. **Data Structure**:
    - Use an array to store the list of products.
    - Each product has the following attributes:
        - productName (string): The name of the product.
        - quantity (integer): The quantity of the product in stock.
        - price (float): The price of the product.
2. **Functions**:
    - **addProduct(productName, price)**: Adds a new product to the inventory with a starting quantity of 0.
    - **updateQuantity(index, quantity)**: Updates the quantity of the product at the specified index.
    - **sellProduct(index, quantity)**: Decreases the quantity of a product by the specified quantity when the product is sold
    - **displayInventory()**: Displays all products with their names, quantities, and prices.
    - **calculateTotalValue()**: Calculates and displays the total value of all products in stock (quantity * price).
3. **Program Flow**:
    - Users can add products using the addProduct function.
    - Users can update the quantity of a product using the updateQuantity function.
    - Users can sell products, which reduces the quantity in stock, using the sellProduct function.
    - The system allows users to view the current inventory using the displayInventory function.
    - The total value of the inventory is calculated and displayed using the calculateTotalValue function.

Output

```
Product added: Laptop, Price: 1000.00
Product added: Mouse, Price: 25.50
Product added: Keyboard, Price: 45.75

Updated quantity of Laptop: 10
Updated quantity of Mouse: 50
Updated quantity of Keyboard: 30

Sold 5 units of Laptop. Remaining stock: 5
Sold 10 units of Mouse. Remaining stock: 40

Displaying Inventory:
1. Laptop, Quantity: 5, Price: 1000.00
2. Mouse, Quantity: 40, Price: 25.50
3. Keyboard, Quantity: 30, Price: 45.75

Total value of inventory: 17610.00
```

Q2: **[wgt: 15]**

You are required to create a **tourism-themed webpage** using HTML and inline CSS. First, create a file called tourism_exploration.html with a complete HTML structure and title the page as **"Discover New Destinations!"** The page should contain several sections, each with specific styles applied directly through inline CSS. The webpage should include the following:

1. **Hero Section**: The H1 heading should read **"Discover New Destinations!"** with the following style:
   - Light blue text
   - Font size of 48px
   - Center-aligned text
   - 80px top margin
   - A background image should be used to enhance the look, covering the entire section.
2. **Navigation Bar:** The navigation bar should contain several anchor links to different sections on the page (e.g., Why Travel? Top Destinations, Travel Tips, Connect, Login). Ensure the following:
   - Each link should have a 15px margin.
   - Use a dark blue background for the navbar.
   - The **Login button** should be highlighted in bold with a background color of orange (#ff8000) and padding of 10px 20px. It should be positioned on the right side of the navbar.
   - Use inline CSS to style the links with appropriate font sizes (18px) and colors (white).
3. **Why Travel Section:**
   - The H2 heading should read **"Why Travel?"** with dark blue text, a font size of 36px, center-aligned text, and a 20px bottom margin.
   - The section should have a background color of light gray (#f9f9f9) with 50px padding around the content.
4. **Top Destinations Section:**
   - The H3 heading should read **"Top Destinations to Explore"** with dark green text, a font size of 32px, and italicized text.
   - Below the heading, provide **3 destination images** (e.g., Paris, Tokyo, Sydney). Each image should:
     - Have a border of 5px solid dark green.
     - A border-radius of 10px.
     - Be linked to a relevant webpage/resource about the destination.
     - The images should be displayed side-by-side in a responsive manner with a 20px gap.
   - Each image should have a caption under it that says the name of the location, with the text centered and font size of 18px.
5. **Travel Tips Section:**

- The H4 heading should read **"Essential Travel Tips"** with purple text, a font size of 30px, and bold text.
- Below the heading, provide a list of 5 travel tips using an unordered list:
  - Each list item should be styled with a small icon (use emojis or placeholder icons).
  - The font size for the list items should be 18px with 10px margin at the bottom.

6. **Connect Section:**
   - The H5 heading should read **"Connect with Fellow Travelers"** with dark red text, a font size of 28px, center-aligned text, and a yellow background.
   - Below the heading, provide a **link to a travel community resource** (e.g., Lonely Planet) with a relevant image (e.g., a group of travelers).
   - The image should have a solid border and padding of 10px.
7. **Login Section:**
   - The H6 heading should read **"Ready to Start Your Journey?"** with dark blue text, a font size of 26px, and centered alignment.
   - Below the heading, provide a **Login button** styled as a link with a background color of orange (#ff8000), white text, and padding of 10px 20px. The button should be clearly visible and placed at the center of the section.
   - The button should be an anchor link that directs the user to a login page (e.g., https://example.com/login).
8. **Footer:**
   - The footer should contain a copyright notice **"© 2024 Discover New Destinations. All rights reserved."** styled with white text and a background color of dark blue.
   - The font size should be 16px, and the text should be centered.

[wgt: 20]

Q3:

You are tasked with creating an **Employee Management System** using **Object-Oriented Programming (OOP) principles.** The system will manage two types of employees: **Full-time** and **Part-time** employees. The program should use **classes** and support **inheritance** and **polymorphism.**

Requirements:

1. **Classes and Data Members:**
   - **Base Class: Employee**
     - Attributes: name, id, department, and salary.
     - A **virtual function** calculateSalary() to be overridden in derived classes.
     - A **function** displayDetails() to print employee details.
   - **Derived Class: FullTimeEmployee** (inherits from Employee)
     - Additional attribute: bonus (for the bonus of full-time employees).
     - Overriding calculateSalary() to include a fixed salary plus bonus.
   - **Derived Class: PartTimeEmployee** (inherits from Employee)
     - Additional attributes: hourlyRate and hoursWorked.
     - Overriding calculateSalary() to calculate salary based on hours worked.

2. **Functions:**

- **addFullTimeEmployee():** Adds a full-time employee by collecting input for name, id, department, and bonus.
- **addPartTimeEmployee():** Adds a part-time employee by collecting input for name, id, department, hourly rate, and hours worked.
- **displayAllEmployees():** Displays the details of all employees, including their salary calculated by the respective calculateSalary() method.

3. **Program Flow:**

- Use **inheritance** to define common and specific properties for full-time and part-time employees.
- Use **polymorphism** to calculate and display salaries for different employee types.
- Store the employees in an array and allow the user to interactively add employees and view the details.
-

Output

```
Employee Management System
1. Add Full-Time Employee
2. Add Part-Time Employee
3. Display All Employees
4. Exit
Enter your choice: 1
Enter Full-Time Employee Details:
Name: John
ID: 101
Department: IT
Bonus: [illegible]

Employee Management System
1. Add Full-Time Employee
2. Add Part-Time Employee
3. Display All Employees
4. Exit
Enter your choice: 2
Enter Part-Time Employee Details:
Name: Jane
ID: 102
Department: HR
Hourly Rate: [illegible]
Hours Worked: 80

Employee Management System
1. Add Full-Time Employee
2. Add Part-Time Employee
3. Display All Employees
4. Exit
Enter your choice: 3

Employee Details:
Name: John, ID: 101, Department: IT, Salary: [illegible]
Bonus: [illegible], Total Salary: [illegible]
--------------------
Name: Jane, ID: 102, Department: HR, Salary: [illegible]
Hourly Rate: [illegible], Hours Worked: 80
-------------------- [illegible]
```

<<<<<<<<<< Best of Luck :) >>>>>>>>>>