

Computer Programming Examination

Beginner Level: Python & C Language

Total Duration: 1 Hour 45 Minutes | **Total Marks:** 40

INSTRUCTIONS:

1. Answer both questions.
 2. Ensure your code is well-commented.
 3. Strictly follow the functional requirements.
-

Section A: Python Programming

Duration: 45 Minutes | **Marks:** 20

Question 1: The 'ClimateView' Simulator

Context: You have been hired to build a prototype for a Command Line Interface (CLI) weather application called "ClimateView." Because this is a prototype, you will not connect to a live internet API. Instead, you must simulate weather data using internal data structures.

Objective: Write a complete Python script that simulates a weather application. Your program must demonstrate mastery of: *Dictionaries, Lists, Functions, while Loops, Conditional Logic, and Exception Handling.*

Functional Requirements:

1. **Data Structure (The Database):** Create a global dictionary named `weather_db`. The keys should be city names. The values should be a nested dictionary containing: temperature (int), condition (string), and humidity (int). Pre-populate with 3 cities.
2. **The `get_weather(city)` Function:** Look up the city. If found, print a formatted report. If not found, raise a `ValueError` or print a friendly error.
3. **The `add_city()` Function:** Prompt user for City Name, Temp, Condition, and Humidity. Store this in `weather_db`.
4. **Main Execution Loop:** Use a while loop for the menu (Check Weather, Add City, Exit). Handle non-numeric input gracefully.

Sample Output:

```
=== ClimateView Menu ===
1. Check Weather
2. Add New City
3. Exit
Select an option: 1

Enter city name: KAMPALA

--- Weather Report for Kampala ---
```

Temperature: 28°C
Condition: Sunny
Humidity: 60%

Section B: C Language Programming

Duration: 60 Minutes | Marks: 20

Question 2: The 'TaskMaster' To-Do List

Context: You are developing a task management tool for a CLI environment. The system needs to manage a list of tasks efficiently using C structures and file handling.

Objective: Write a complete C program to manage a To-Do list. Your program must demonstrate mastery of: *Structs, Arrays, Pointers, Functions (Pass-by-value vs Pass-by-reference), Switch Statements, and basic File I/O.*

Functional Requirements:

- The Data Structure:** Define a struct named Task with members: id (int), description (string size 100), and is_completed (int: 0 or 1).
- Global Storage:** Declare an array Task taskList[50] and an integer task_count.
- Required Functions:**
 - void addTask(...): Assign ID, get description, set default status.
 - void viewTasks(...): Loop and print tasks with status [x] or [].
 - void markComplete(...): Update status to 1 based on ID.
 - void saveToFile(...): Write tasks to 'tasks.txt'.
- Menu Logic:** Loop in main() with a switch statement for: Add, View, Mark Complete, Save & Exit.

Sample Output:

```
=== TaskMaster v1.0 ===
1. Add Task
2. View All Tasks
3. Mark Task as Complete
4. Save and Exit
Choice: 2

ID | Status | Description
-----
1  | [ ]    | Finish C Assignment
2  | [x]    | Buy Groceries
-----
```

End of Examination Paper