FORMATIVE ASSESSMENT 2 (Project)

Programming With Python

PRP411

2024

CTU training solutions

Creative & Technology Universitas

Table of Contents

Please complete the declaration of authenticity below for all assignments:

DECLARATION OF AUTHENTICITY

I ____ hereby

(FULL NAME)

declare that the contents of this assignment are entirely my own work with the exception of the following elements: (List the elements of work in this project that were not self-generated as well as who the originator of the element is)

Element	Originator

Signature: ______________________________ Date: _____________

Programming Foundation

Module: Programming with Python

Module Code: PRP411

NQF: 4

Credits: 21

Assessment type: Formative Assessment

Mark allocation: 100

Hand out date: 05 April 2024

Hand in date: 3 May 2024

Instructions:

Read each question carefully and consider the mark allocation prior to answering.

- Ensure you answer all questions.
- The use of AI tool to complete this assessment is strictly prohibited.
- For final submission**,** include the Declaration of Authenticity **and source code**

Unit Standards:

- Apply the principles of Computer Programming.
- Design a computer program according to the given specifications.
- Write/present/sign texts for a range of communicative contexts.

Unit Standards:

- Nested if Statements.
- Nested loops.
- Looping Through a Dictionary.
- Functions.

Project Questions (100 Marks)

Instructions: Choose the correct option for each question below.

Event Registration System.

Events management involves planning, executing and evaluating different types of events, including concerts, conferences, weddings, product launches, and sports tournaments. Events management is a fast-growing and fast-paced industry that involves activities such as:

- Identifying and securing venues for events.
- Planning event schedules.

Events management is both exciting and stressful: it gives you the opportunity to work with different types of people, to express your creativity, and to plan different types of events, but it also requires you to work under pressure, to meet deadlines, and to meet (or exceed) client expectations.

Source: www.oxbridgeacademy.edu.za

Features to Implement

- **Storing Event Information:**
  Use a dictionary to maintain event information. Each event should include an ID, name, maximum attendees, and a list of registered attendees.

- **Adding Events:**
  Create a function to add new events to the system. This should include obtaining event details from the user and updating the event collection.

- **Registering for Events:**
  Implement a function that allows users to register for an event by event ID. Ensure that registration is only successful if the event has not reached its maximum attendees.

- **Listing Events:**
  Define a function that lists all events, showing their ID, name, and the number of available spots.

- **Viewing Event Registrations:**
  Create a function to view all attendees registered for a specific event. This requires iterating through the event's attendees list.

- **Search Functionality:**
  Implement a function allowing users to search for events by name. Use nested loops and if statements to search through events and find matches.

Tasks

1. **Storing Event Information**
   Initialize a dictionary named `events` to store the event information. Each key in the dictionary is an event ID, and its value is another dictionary with `name`, `max_attendees`, and `attendees` (a list).

2. **Adding Events**
   Define a function `add_event()` that asks the user for event details (name and maximum attendees) and adds a new event to the `events` dictionary.

3. **Registering for Events**
   Define a function `register_for_event(event_id, attendee_name)` that checks if the event has space. If so, add the attendee to the event's attendee list. Use nested if statements to validate the event ID and check for availability.

4. **Listing Events**
   Implement a function `list_events()` that displays all events, including their ID, name, and how many spots are left.

5. **Viewing Event Registrations**
   Define a function `view_event_attendees(event_id)` to list all attendees of a specific event. Validate the event ID and then iterate through the list of attendees.

6. **Search Functionality**
   Implement a function `search_events(keyword)` to allow searching events by name. Loop through the events dictionary, using nested if statements to match the keyword against event names.

End of Project Questions **[Total = 100 Marks]**

End of paper.

Formative Assessment 2 Rubric

Penalties: AI-generated code snippets and any form of duplicate, plagiarism and/or assisted tasks to complete this assessment will result in a final grade of zero (0). Additional penalties will be applied for incomplete logic on the program.

Formative Assessment 2		
Student name/number:		
Criterion	**Mark Allocation**	
	10 – 6	**5 - 0**
Functionality and Completeness (40 Marks)		
Event Adding Functionality (10 Marks): Correct implementation of the function to add new events, including accurate updates to the event collection.		
Event Registration Functionality (10 Marks): Proper implementation of the registration function, ensuring it checks for maximum attendees and updates the event's attendees list accordingly.		
List and View Events (10 Marks): Accurate display of events and their details (ID, name, spots left) and the ability to view all attendees of a specific event.		
Search Functionality (10 Marks): Effective implementation of the search functionality, allowing users to find events by name.		
Code Quality and Style (20 Marks)		
Readability (10 Marks): Code is well-organized, with clear variable names and comments explaining the purpose and logic of each section/function.		
Efficiency (10 Marks): Efficient use of data structures and control structures, minimizing redundancy and optimizing for performance where possible.		
Correct Use of Data Structures (20 Marks)		
Appropriate Data Structures (10 Marks): Effective use of dictionaries and lists to store and manage event and registration information.		
Data Structure Operations (10 Marks): Correct implementation of operations on data structures, including additions, updates, and lookups.		
Criterion	**Mark Allocation**	
	5 - 3	**2 - 0**
Error Handling and Validation (10 Marks)		
Input Validation (5 Marks): Proper validation of user inputs to prevent errors due to invalid data.		
Error Handling (5 Marks): Implementation of error handling mechanisms to manage exceptions and provide useful feedback to the user.		
Documentation and Presentation (10 Marks)		
Code Documentation (5 Marks): Inclusion of comprehensive comments and documentation within the code, explaining the functionality and logic.		
Project Presentation (5 Marks): Clarity in presenting the project, with a well-structured .py file that follows a logical flow and is easy to navigate.		
TOTAL		/ 100