

1. What is the primary goal of Object-Oriented Programming (OOP)?
2. What is an object in Python?
3. What is a class in Python?
4. What are attributes and methods in a class?
5. What is the difference between class variables and instance variables in Python?
6. What is the purpose of the self parameter in Python class methods?
7. For a library management system, you have to design the "Book" class with OOP principles in mind. The "Book" class will have following attributes:
 - a. title: Represents the title of the book.
 - b. author: Represents the author(s) of the book.
 - c. isbn: Represents the ISBN (International Standard Book Number) of the book.
 - d. publication_year: Represents the year of publication of the book.
 - e. available_copies: Represents the number of copies available for checkout.

The class will also include the following methods:

- a. check_out(self): Decrements the available copies by one if there are copies available for checkout.
 - b. return_book(self): Increments the available copies by one when a book is returned.
 - c. display_book_info(self): Displays the information about the book, including its attributes and the number of available copies.
8. For a ticket booking system, you have to design the "Ticket" class with OOP principles in mind. The "Ticket" class should have the following attributes:
 - a. ticket_id: Represents the unique identifier for the ticket.
 - b. event_name: Represents the name of the event.
 - c. event_date: Represents the date of the event.
 - d. venue: Represents the venue of the event.
 - e. seat_number: Represents the seat number associated with the ticket.
 - f. price: Represents the price of the ticket.
 - g. is_reserved: Represents the reservation status of the ticket.

The class also includes the following methods:

- a. reserve_ticket(self): Marks the ticket as reserved if it is not already reserved.
 - b. cancel_reservation(self): Cancels the reservation of the ticket if it is already reserved.
 - c. display_ticket_info(self): Displays the information about the ticket, including its attributes and reservation status.
9. You are creating a shopping cart for an e-commerce website. Using OOP to model the "ShoppingCart" functionality the class should contain following attributes and methods:
 - a. items: Represents the list of items in the shopping cart.

The class also includes the following methods:

- a. `add_item(self, item)`: Adds an item to the shopping cart by appending it to the list of items.
- b. `remove_item(self, item)`: Removes an item from the shopping cart if it exists in the list.
- c. `view_cart(self)`: Displays the items currently present in the shopping cart.
- d. `clear_cart(self)`: Clears all items from the shopping cart by reassigning an empty list to the items attribute.

10. Imagine a school management system. You have to design the "Student" class using OOP concepts. The "Student" class has the following attributes:

- a. `name`: Represents the name of the student.
- b. `age`: Represents the age of the student.
- c. `grade`: Represents the grade or class of the student.
- d. `student_id`: Represents the unique identifier for the student.
- e. `attendance`: Represents the attendance record of the student.

The class should also include the following methods:

- a. `update_attendance(self, date, status)`: Updates the attendance record of the student for a given date with the provided status (e.g., present or absent).
- b. `get_attendance(self)`: Returns the attendance record of the student.
- c. `get_average_attendance(self)`: Calculates and returns the average attendance percentage of the student based on their attendance record.

Submission Guidelines:

- Answer all the questions in a single Jupyter Notebook file (.ipynb).
- Include necessary code, comments, and explanations to support your answers and implementation.
- Ensure the notebook's cells containing code are already run.
- Create a GitHub repository to host your assignment files.
- Rename the Jupyter Notebook file using the format "date_month_topic.ipynb" (e.g., "01_July_OOPs.ipynb").
- Place the Jupyter Notebook file in the repository.
- Ensure the repository is publicly accessible.
- Submit the link of the corresponding assignment present in your GitHub repository as the assignment submission link.

Note:- Create your assignment in Jupyter notebook and upload it to GitHub & share that uploaded assignment file link through your dashboard. Make sure the repository is public.