OOPs

1. **What is the primary goal of Object-Oriented Programming (OOP)?**

Its primary goal is to create a structured program which is simple and reusable. Reusing is by child class can use properties of parent class

1. **What is an object in Python?**

A object is instance of class. It is copy of class with actual values. Any entity with a state and behavior is an object

1. **What is a class in Python?**

A class is blue print of the object. It is template for object.

1. **What are attributes and methods in a class?**

A variable stored in an instance or class is attribute. A function stored in an instance or class is method.

1. **What is the difference between class variables and instance variables in Python?**

Class variable can be used to store data for all instance/object of class but instance variable do not have direct control on other instance of the class

1. **What is the purpose of the self parameter in Python class methods?**

SELF represents the instance of class. This keyword allows you to access variables, attributes, and methods of a defined class in Python. The self parameter doesn't have to be named “self,” as you can call it by any other name

**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.**

Please refer ipynb file

**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.**

Please refer ipynb file

**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.**

Please refer ipynb file

**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.**

Please refer ipynb file