Comprehending Constructor & Destructor

1. Design a class ParkingSlot with attributes slotNumber and isOccupied. Use a default constructor to initialize all parking slots as empty. Write member functions to mark a slot as occupied and display the slot status.
2. Create a class Book with attributes title, author, and isAvailable. Use a default constructor to set all attributes to default values. Add member functions to update availability when a book is borrowed or returned, and display the book details.
3. Implement a class Product with attributes productID, name, and price. Use a parameterized constructor to initialize a product. Add member functions to calculate the total price for a given quantity and display product details.
4. Define a class GymMember with attributes memberID, name, and membershipFee. Use a parameterized constructor to register a new member. Add member functions to calculate and display discounted membership fees for different membership types.
5. Create a class ElectricityBill with attributes consumerNumber, unitsConsumed, and ratePerUnit. Use a constructor with default arguments to initialize the rate per unit to a standard value. Add member functions to calculate and display the total bill amount.
6. Design a class Room with attributes roomNumber, roomType, and rate. Use a constructor with a default argument to initialize the rate based on the room type. Add member functions to calculate the bill for a given number of days and display the booking details.
7. Write a class BankAccount with attributes accountNumber, holderName, and initialBalance. Overload the constructor to allow creating an account with or without an initial balance. Add member functions to perform deposits, withdrawals, and display the account details.
8. Define a class Event with attributes eventID, eventName, and eventDate. Overload constructors to register an event with only the name or with all attributes. Add a member function to check if the event is scheduled within the next month.
9. Create a class Employee with attributes employeeID, name, and salary. Use a copy constructor to duplicate an existing employee's details into a new record. Add member functions to modify and display the duplicated employee's details.
10. Write a class Vehicle with attributes registrationNumber, ownerName, and insuranceStatus. Use a copy constructor to create a duplicate insurance file for the same vehicle. Add member functions to modify insurance details and display the records.

Comprehending Constructor & Destructor

11. Implement a class FileManager with attributes fileName and isOpen. Open the file in the constructor and close it in the destructor. Add member functions to read or write data into the file and display file status.

12. Design a class Connection with attributes connectionID and status. Establish a connection in the constructor and terminate it in the destructor. Add member functions to send data and check connection status.

13. Create a class HotelBooking with attributes bookingID, customerName, and totalAmount.

- Use a default constructor to set initial values for walk-in customers.
- Use a parameterized constructor to initialize for online bookings.
- Use a copy constructor to duplicate a booking for modifications.

  Add member functions to update room preferences and display booking details.

14. Implement a class Flight with attributes flightNumber, destination, and fare.

- Use constructor overloading to initialize a flight with only flightNumber or all attributes.
- Use a destructor to release reservation data when a flight is canceled.

  Add member functions to calculate fare for different ticket types (economy, business).

15. Design a class EWallet with attributes walletID, ownerName, and balance.

- Use a parameterized constructor to initialize the wallet with an initial deposit.
- Use a copy constructor to create a backup of the wallet details.
- Use a destructor to clear sensitive data when the wallet object is destroyed.

  Add member functions to add and withdraw money, and display wallet status.

16. Create a class Student with attributes rollNumber, name, and grade.

- Use a default constructor to initialize students who haven't received grades yet.
- Use a parameterized constructor to initialize students with grades.

  Add member functions to update and display grades.

17. Write a class Payment with attributes transactionID, amount, and status.

- Use constructor overloading to initialize either with transactionID or with all attributes.

- Use a destructor to delete sensitive transaction data after processing.

  Add member functions to check and display payment status.

18. Define a class MovieTicket with attributes ticketID, movieName, and price.

- Use a parameterized constructor to initialize the ticket.
- Use a copy constructor to duplicate the ticket for group bookings.

  Add member functions to calculate and display the total cost for multiple tickets.

19. Design a class Item with attributes itemCode, name, and quantity.

- Use constructor overloading to initialize an item with either just the name or all attributes.

  Add member functions to update stock and display inventory details.

20. Implement a class Patient with attributes patientID, name, and disease.

- Use a default constructor to register walk-in patients without initial diagnosis.
- Use a parameterized constructor for patients with a known disease.
- Use a copy constructor to transfer patient details to a referral system.

  Add member functions to update and display patient details.