1. Write a program to create class Student with student's rollno, name and marks of three subjects (OOCP, DBMS and English) and display the details of student with total marks of all subjects along with the percentage in proper format.(Output should be in descending order of percentage).

2. Design a class hierarchy for different types of bank accounts (Savings, Current).

   Use constructors to initialize account details.

   Use operator overloading (+, -) to perform deposit and withdrawal.

   Implement a template class for transaction logs that can store any type of transaction (int, float, string).

   Apply inheritance so that Savings and Current accounts extend a base Account class.

3. Create a class EMPLOYEE with data members empId, name, and salary. Initialize salary with a default value. Using a member function, update the salary of an employee for a given empId entered by the user.

4. Create a class for products with attributes like name, price, and quantity.

   Use constructors to initialize product details.

   Overload operators (+ to merge carts, == to compare products).

   Use a template class to manage cart items of any type (physical, digital).

   Use inheritance for specialized product categories like Electronics, Clothing.

5. Create a class BANK having account number, name, and balance as data members. Provide functions to deposit and withdraw money. Update the balance based on user choice and display final account details.

6. Build a system to manage books and members.

   A Book base class with constructors for title, author, ISBN.

   Overload operators (++ to increment available copies, -- to decrement).

   A template class for managing lists of books or members.

   Inheritance: EBook and PrintedBook classes derived from Book.

7. Create a class BOOK with data members isbn, title, and price. Initialize book details using a constructor. Using a member function, update the price of the book when the user provides the isbn number.

8. Create a class hierarchy for Student and GraduateStudent.

   Constructors should initialize student info and marks.

   Overload operators (+ to calculate combined GPA, < to compare).

   Use a template class Result<T> to store marks in int, float, or double.

   Use inheritance so specialized types of students inherit from the base Student.

9. Create a class PRODUCT with product id, product name, and stock quantity as data members. Initialize stock with a given value. Using a member function, allow the user to update the stock for a specific product id.
10. Build a base class Order with derived classes DineIn and TakeAway.

   Constructors for order details (items, price).

   Overload operators (+ to add orders, - for discounts).

   A template class for storing menu items of different types.

   Inheritance to differentiate order categories.

11. Create a class TEACHER with teacher id, name, and subject as data members. Using a member function, update the subject taught by the teacher when the teacher id is entered by the user.

12. Create a base class Vehicle.

Constructors for vehicle details (model, rent per day).

Overload operators (+ to extend rental days, < to compare rent).

Template class Rental<T> to store vehicle data in different formats.

Derived classes: Car, Bike, Truck.

13. Create a class VEHICLE with registration number, model, and mileage as data members. Initialize values using a constructor. Provide a member function to update mileage for a vehicle when the registration number is given.

14. Create a base class Ticket with details like passenger name, seat number, and fare.

Use constructors to initialize ticket info.

Overload operators (+ to merge multiple tickets for group booking, == to compare seat numbers).

Use a template class Booking<T> to handle booking IDs (int, string, etc.).

Inheritance for DomesticTicket and InternationalTicket.

15. Create a class MOVIE with movie id, title, and rating as data members. Initialize rating with a default value. Using a member function, update the rating of a movie when the user provides the movie id.

16. Base class Exam with attributes like subject, duration, and max marks.

Constructors to initialize exam details.

Overload operators (+ to add marks of two subjects, < to compare scores).

Template class AnswerSheet<T> to handle answers in text, numbers, or MCQs.

Inheritance for OnlineExam and OfflineExam.

17. Create a class COURSE with course id, course name, and duration (in months) as data members. Initialize using a constructor. Using a member function, update the duration when the user provides the course id.

18. Base class Medicine with attributes: name, price, expiry date.
Constructors for initialization.
Overload operators (* for total cost = price × quantity, == to check same medicine).
Template class Inventory<T> for managing medicines by batch ID (int, string).
Inheritance: Tablet, Syrup, Injection from Medicine.

19. Write a program to create class Time (int h, int m, int s). Read a value as seconds from user to display new time after adding the value to minutes and hours in Time.

20. Write a program to count number of objects of a class employee having id, name, basic salary. Calculate the average gross salary of all employee(Take at least three employee information). (HRA 20%, DA 30%)

21. Write a program to define a class called book. Write a program to read information about 10 books and display books details in ascending order of price in proper format.

22. WAP to define a matrix class and overload the * operator to multiply a number with matrix (Example: 6*Matrix should be possible. Take number from user).

23. Create class STUDENT having rollno, name and age as data members, also take subject with three subjects and initialize their value with minimum passing marks. Using member function, modify marks of student with specific rollno which is given by user.

24. Write a function called find(). find() searches an array for an employee. It returns either the index of the matching object(Match using id of employee) (if one is found) or -1 if no match is found. If match found display the details.

25. Write a program to create class Student with student's rollno, name and marks of three subjects (OOCP, DBMS and English) and display the details of student with total marks of all subjects along with the percentage in proper format.(Output should be in descending order of percentage).

26. WAP to define a class Date with properties int month; int day; int year; Overload the + operator [a+b] (a is of date type and b is an integer – No of days to add in date), use the assumption that all years have 360 days and months 30 days.