1. Create a class `Car` with:

- private `speed`
- function `setSpeed(int)`
- **const function** `showSpeed()`

---

2. Create a class `Book` with:

- title
- price

Create **array of 5 books** and display them using loop.

---

3. Create a class `BankAccount` with:

- static float `interestRate`
- static function `changeRate(float)`
- function `showRate()`

Change interest rate once and show for all objects.

---

4. Create two classes:

Engine

- horsepower

Car

- model
- Engine object

Print car details with engine power.

---

5. Scenario:
   A university wants to track students and departments.

Create:

- `Department` class (name, code)
- `Student` class (name, age, Department object)

Create **array of 3 students** and display all data.

---

6. A game server tracks total players online.

Create class `Player`:

- name
- static `totalPlayers`
- constructor increments players
- destructor decrements players
- static function `showPlayers()`

Test by creating objects inside a block `{ }`.

---

7. Bank wants to track accounts and interest policy.

Class `Account`:

- accountHolder
- balance
- static `bankName`
- static `interestRate`
- const function `showBalance()`
- function `deposit()`

Create **3 accounts** and demonstrate:

- deposit
- static value sharing
- const function usage

8. An online game tracks players and their weapons.

Requirements:

- Create a `Weapon` class:
    - `weaponName`
    - `damage`
- Create a `Player` class:
    - `playerName`

- o `health`
  - o **Weapon object inside Player (Composition)**
- Use **array of Player objects**
- Use **static int activePlayers**
- Create a **const function** `showStatus()`
- When a player object is destroyed, active players should reduce

📌 **Hint:** Weapon should not exist without Player.

9. A software company manages employees and projects.

Requirements:

- Create a `Project` class:
  - o `projectTitle`
  - o `duration`
- Create an `Employee` class:
  - o `empName`
  - o `salary`
  - o **Has-A relationship with Project (Aggregation)**
- Store **multiple employees in an array**
- Use **static float taxRate** shared by all employees
- Use **const function** `calculateNetSalary()`
- Use **static function** to change tax rate

📌 **Hint:** One project can exist even if employee leaves.

10. A bank manages customer accounts and branches.

Requirements:

- Create a `Branch` class:
  - o `branchName`
  - o `city`
- Create an `Account` class:
  - o `accountHolder`
  - o `balance`
  - o **Has-A relationship with Branch (Aggregation)**
- Use **array of Account objects**
- Use **static string bankName**
- Use **static float interestRate**
- Use **const function** `showBalance()`
- Use **static function** to update interest rate for all accounts

📌 **Hint:** Branch exists even if accounts are closed.