📘 JavaScript OOP & ES6 Assignments

🟢 PART 1: Classes & Objects

Assignment 1: Create a Basic Class (Beginner)

🎯 Objective:

Understand class structure, constructor, and object creation.

📝 Task:

Create a class called Person.

Requirements:

- Properties: name, age
- Method: greet() → prints
  "Hello, my name is <name> and I am <age> years old."

✅ Expected Usage:

```
let p1 = new Person("Rahul", 25);

p1.greet();
```

Assignment 2: Bank Account System

🎯 Objective:

Use methods and modify object state.

📝 Task:

Create a class BankAccount.

Properties:

- accountHolder
- balance

Methods:

- deposit(amount)
- withdraw(amount)
- checkBalance()

Conditions:

- Cannot withdraw more than balance
- Show appropriate message

Assignment 3: Student Grade Calculator

🎯 **Objective:**

Work with arrays inside classes.

📝 **Task:**

Create class Student

Properties:

- name
- marks (array)

Methods:

- addMark(mark)
- getAverage()
- getGrade()

Grade Rules:

- 90+ → A
- 75+ → B
- 50+ → C
- else → Fail

🟡 PART 2: Inheritance

Assignment 4: Vehicle Inheritance

🎯 **Objective:**

Understand extends and super.

📝 **Task:**

Create:

Base Class:

Vehicle

- brand

- speed

- start()

Derived Class:

Car

- fuelType

- showDetails()

Requirements:

- Use super() properly.
- Call parent method from child object.

Assignment 5: Employee Salary System

🎯 **Objective:**

Multi-level inheritance

Structure:

Employee

|

```
Manager
  |
Director
```

Base Class:

- name
- salary
- getDetails()

Manager:

- bonus
- getTotalSalary()

Director:

- stockOptions
- getFullCompensation()

Assignment 6: Shape Area Calculator

Base Class:

Shape

- calculateArea()

Child Classes:

- Circle
- Rectangle
- Triangle

Each class must override calculateArea().

Test polymorphism using:

let shapes = [new Circle(...), new Rectangle(...)]

Loop through and call calculateArea().

🔵 PART 3: Polymorphism

Assignment 7: Method Overriding

Create:

```
class Animal {
  makeSound()
}
```

Child Classes:

- Dog
- Cat
- Cow

Each should override makeSound() differently.

Test using:

```
let animals = [new Dog(), new Cat(), new Cow()];
animals.forEach(a => a.makeSound());
```

Assignment 8: Runtime Polymorphism Project

Create a Payment system.

Base Class:

Payment

- pay(amount)

Child Classes:

- CreditCardPayment
- UPIPayment
- CashPayment

Each should implement pay() differently.

🟣 PART 4: ES6 New Features (With OOP)

Assignment 9: Use ES6 Features Inside Class

Create a Product class using:

- Default parameters
- Template literals
- Arrow functions
- Destructuring
- Spread operator

Example:

constructor({name, price, category = "General"})

Assignment 10: Static Methods

Create a class MathUtils

Add:

- static add()
- static subtract()
- static multiply()

Test:

MathUtils.add(10, 20);

Assignment 11: Getters & Setters

Create class User

- Private property _password

- Setter should validate:
  - Minimum 6 characters
- Getter should return full password

Assignment 12: Private Fields (ES2022)

Use:

#balance

Create class Wallet

Methods:

- addMoney()
- spendMoney()
- getBalance()

Ensure balance cannot be accessed directly.