## **Assignment 1: Builder Pattern - Configurable Computer Assembly**

### **Objective:**

Students will implement the **Builder Pattern** to construct a computer with different configurations.

### **Problem Statement:**

A computer store wants to allow customers to configure their PCs before purchase. The store sells computers with different processors, RAM, storage, and GPUs. Each computer must have:

* A **mandatory** motherboard and power supply.
* **Optional** components like GPU, additional storage, and cooling system.

### **Requirements:**

1. Implement a **ComputerBuilder** class that allows step-by-step computer customization.
2. Provide multiple configurations such as:
   * **Gaming PC** (High-end CPU, GPU, SSD, Liquid Cooling).
   * **Office PC** (Mid-range CPU, SSD, No GPU).
   * **Server** (Multiple HDDs, High RAM).
3. Ensure that the **Computer** object is immutable after it has been built.
4. Provide a Director class to create predefined builds.

### **Expected Output:**

Students will implement a ComputerBuilder class and demonstrate how to create various configurations.

## **Assignment 2: Factory Pattern - Employee Management System**

### **Objective:**

Students will implement the **Factory Pattern** to generate different types of employees in a company.

### **Problem Statement:**

A company has different types of employees: **Full-Time, Part-Time, and Interns**. Each type has different salary structures and benefits.

### **Requirements:**

1. Create an abstract base class **Employee** with the following:
   * calculateSalary()
   * getBenefits()
2. Implement concrete classes:
   * **FullTimeEmployee**
   * **PartTimeEmployee**
   * **Intern**
3. Implement a **Factory Method** (EmployeeFactory) to create employee objects based on input parameters.
4. Use the Factory in a **Payroll System** to generate employee details dynamically.

### **Expected Output:**

Students will implement a EmployeeFactory class that generates employees based on user input.