# TEST

## Question 1: Book Management System with Switch Case Statement

**Task:**
Create a simple **Book Management System** in Java using a **switch-case** statement. The system should provide the following options:

1. Add a book (store book details like title and author).
2. Display all books.
3. Search for a book by title.
4. Exit the program.

- Use an **ArrayList** to store book details.
- Implement a loop to keep the program running until the user chooses to exit.

---

## Question 2: Method Overloading and Method Overriding

**Task:**
Write a Java program to demonstrate **Method Overloading and Method Overriding**.

- **Method Overloading:** Create a `Calculator` class with multiple `add()` methods that accept different numbers of parameters (two or three integers).
- **Method Overriding:** Create a base class `Animal` with a `makeSound()` method, and a subclass `Dog` that overrides `makeSound()` to provide a specific implementation.
- In the `main()` method, demonstrate both method overloading and method overriding.

---

## Question 3: Interfaces

**Task:**
Create a Java program that defines an interface `Vehicle` with methods:

- `start()`, `stop()`, and `fuelType()`.

Then, implement this interface in two classes:

1. `Car` (returns "Petrol" as fuel type)
2. `ElectricBike` (returns "Electric" as fuel type)

- In the `main()` method, create objects of `Car` and `ElectricBike`, call the methods, and display the outputs.

---

## Question 4: Exception Handling

**Task:**
Write a Java program that demonstrates exception handling using a **try-catch-finally** block.

- Create a method that **divides two numbers**.
- Accept two numbers from the user as input.
- Handle the `ArithmeticException` if the denominator is zero, and print an appropriate error message.
- Ensure that the **finally block** executes, displaying `"End of program."`

---

## Question 5: Throw and Throws

**Task:**
Write a Java program that demonstrates the **throw** and **throws** keywords.

- Create a method `checkAge(int age)` that:
  - Throws an `IllegalArgumentException` if the age is less than 18.
  - Prints `"Eligible to vote"` if age is 18 or above.
- In the `main()` method:
  - Call `checkAge()` and handle the exception using a `try-catch` block.

## Question: Basic Operations on List Interface

**Task:**
Write a simple Java program to demonstrate the basic operations of the **List interface** using an `ArrayList`.

- Create an `ArrayList` of type `String` to store **names of fruits**.
- Perform the following operations:
  1. **Add** five fruit names to the list.
  2. **Print** the list.
  3. **Remove** the second element from the list.
  4. **Update** the last element to a different fruit name.
  5. **Print** the modified list.