1. Create a `Person` class in JavaScript that has the following properties and methods:

1. **Properties:**
    - `name` (string)
    - `age` (number)
2. **Methods:**
    - `introduce()` - This method should return a string in the format: `"Hello, my name is [name] and I am [age] years old."`
3. **Create an instance of the `Person` class** with the name "Alice" and age 30, and use the `introduce()` method to log the introduction to the console.

2. Create a `Rectangle` class in JavaScript with the following properties and methods:

1. **Properties:**
    - `width` (number)
    - `height` (number)
2. **Methods:**
    - `area()` - This method should return the area of the rectangle (i.e., `width * height`).
    - `perimeter()` - This method should return the perimeter of the rectangle (i.e., `2 * (width + height)`).
3. **Create an instance of the `Rectangle` class** with a width of 10 and a height of 5. Use the `area()` and `perimeter()` methods to log both the area and the perimeter to the console

3. Create a `BankAccount` class in JavaScript with the following properties and methods:

1. **Properties:**
    - `accountHolder` (string)
    - `balance` (number)
2. **Methods:**
    - `deposit(amount)` - This method should add the given amount to the balance.

    - `withdraw(amount)` - This method should subtract the given amount from the balance. Ensure that the balance does not go below zero.
    - `getBalance()` - This method should return the current balance.
3. **Create an instance of the `BankAccount` class** with the account holder's name as "John Doe" and an initial balance of 1000. Perform the following actions:
    - Deposit 500 into the account.
    - Withdraw 200 from the account.
    - Withdraw 1500 from the account (this should not be allowed, and the balance should remain unchanged).
    - Use the `getBalance()` method to log the final balance to the console.

4. Create a `Car` class in JavaScript with the following properties and methods:

1. **Properties:**
    - `make` (string)
    - `model` (string)
    - `year` (number)
    - `mileage` (number)
2. **Methods:**
    - `drive(distance)` - This method should add the given distance to the car's mileage.
    - `getInfo()` - This method should return a string with the car's make, model, year, and mileage in the format: `"Make: [make], Model: [model], Year: [year], Mileage: [mileage]"`.
3. **Create an instance of the `Car` class** with the make "Toyota", model "Camry", year 2020, and an initial mileage of 15000. Perform the following actions:
    - Drive the car for 300 miles.
    - Drive the car for another 150 miles.
    - Use the `getInfo()` method to log the car's information to the console.

5. Create a `Library` class in JavaScript with the following properties and methods:

1. **Properties:**
    - `name` (string)

    - `books` (array) - This will be an array of book objects, where each book object has a `title` (string) and an `author` (string).
2. **Methods:**
    - `addBook(title, author)` - This method should add a new book to the `books` array.
    - `removeBook(title)` - This method should remove a book with the given title from the `books` array. If the book is not found, display a message saying `"Book not found"`.
    - `listBooks()` - This method should return a string listing all the books in the format: `"Title: [title], Author: [author]"`. If there are no books, it should return `"No books available"`.
3. **Create an instance of the `Library` class** with the name "City Library". Perform the following actions:
    - Add the book "The Great Gatsby" by "F. Scott Fitzgerald".
    - Add the book "To Kill a Mockingbird" by "Harper Lee".
    - Remove the book "The Great Gatsby".
    - Use the `listBooks()` method to log the list of books to the console.