

Tasks:

- 1. Write a function `createCounter` that returns another function. The returned function should increment a counter each time**
- 2. Create a recursive function `factorial` that takes a number `n` and returns its factorial. Test the function with various values**
- 3. Write a function `memoizedFactorial` that calculates the factorial of a number and caches results to improve performance on repeated calls**
- 4. Write a function constructor `Person` that takes name and age as parameters and assigns them to the instance. Create an instance of `Person` and log the properties**
4.2. add a method `greet` to the `Person` constructor that logs a greeting message including the person's name. Test this by calling `greet` on an instance of `Person`.
- 4.3. Refactor the `Person` constructor by moving the `greet` method to `Person.prototype`. Create multiple instances and test that they all share the same `greet` method (i.e., it doesn't duplicate for each instance).**

- 5. Write a factory function createUser that can generate Admin and Guest users based on a configuration object. Each type of user should have unique methods (admin with manageUsers, guest with viewContent).**

Creating an Admin user

```
var adminUser = createUser({ type: 'Admin', name: 'Alice' });
```

```
adminUser.manageUsers();
```

Expected output: "Alice is managing users."

Creating a Guest user

```
var guestUser = createUser({ type: 'Guest', name: 'Bob' });
```

```
guestUser.viewContent();
```

Expected output: "Bob is viewing content."