**Introduction:**

In this lab, you will explore **object-oriented programming (OOP)** concepts in JavaScript. You'll create classes, extend them with inheritance, and instantiate objects with unique properties and methods. By the end of this lab, you'll understand how to use OOP to structure and organize your code effectively.

**Goal:**

The goal of this lab is to help learners grasp the principles of object-oriented programming, including class creation, inheritance, and object instantiation.

**Objectives:**

* Create classes and define constructors with default parameter values.
* Use inheritance to extend functionality from a parent class.
* Add and override methods in subclasses.
* Instantiate objects and assign unique properties to each instance.

**Learner Instructions:**

**Lab Tasks:**

**Task 1:** Code a **Person** Class

**Objective:** Create a **Person** class to represent basic human attributes and actions.

**Steps:**

**Step 1:** Open the **ooprogramming.js** file present under the **LEARN** folder. This is where you will write your object-oriented program.

**Step 2:** Define a **Person** class with the following parameters in the constructor:

* **name** (default: **"Tom"**)
* **age** (default: **20**)
* **energy** (default: **100**)

**Step 3:** Add the following methods to the class:

* **sleep()**: Increases the energy property by **10**.
* **doSomethingFun()**: Decreases the energy property by **10**.

**Task 2:** Code a **Worker** Class

**Objective:** Extend the **Person** class to create a **Worker** class with additional attributes and functionality.

**Steps:**

**Step 1:** Define a **Worker** class that inherits from the **Person** class.

**Step 2:** Add two additional parameters to the **Worker** constructor:

* **xp** (default: **0**)
* **hourlyWage** (default: **10**)

**Step 3:** Add the following method to the **Worker** class:

* **goToWork()**: Increases the **xp** property by **10**.

**Task 3:** Code an **Intern** Object

**Objective:** Instantiate the **Worker** class to create an **intern** object with specific attributes.

**Steps:**

**Step 1:** Inside the **intern()** function, create a new **Worker** object with the following properties:

* **name: "Bob"**
* **age: 21**
* **energy: 110**
* **xp: 0**
* **hourlyWage: 10**

**Step 2:** Call the **goToWork()** method on the **intern** object.

**Step 3:** Return the **intern** object.

**Task 4:** Code a **Manager** Object

**Objective:** Instantiate the **Worker** class to create a **manager** object with specific attributes.

**Steps:**

**Step 1:** Inside the **manager()** function, create a new **Worker** object with the following properties:

* **name: "Alice"**
* **age: 30**
* **energy: 120**
* **xp: 100**
* **hourlyWage: 30**

**Step 2:** Call the **doSomethingFun()** method on the **manager** object.

**Step 3:** Return the **manager** object.

**Step 4:** After successfully modifying the **ooprogramming.js** file, navigate to **File** > **Save** to save changes in the file.

**To execute your modified JavaScript code and verify the function is working:**

* Click the **Run Code** button in the top-right corner (shaped like a triangular "Play" button).

**Key Takeaways:**

* Use the **class** keyword to define a blueprint for objects. Constructors initialize properties with default or custom values.
* Use the **extends** keyword to create a subclass, inheriting all properties and methods from a parent class. Subclasses can have their own unique properties and methods.
* Instantiate classes to create specific objects with unique attributes. Use constructor arguments to customize the properties of each instance.
* Methods define specific behaviours that operate on the object's properties, encapsulating functionality within the class.
* OOP makes code modular, reusable, and easier to understand.

**Final Step: Submit Your Code:**

* Go to **File** > **Save** to ensure your work is saved.
* **Submit your assignment**: Click the "**Submit Assignment"** button in the Lab toolbar.
  + Your code will be **autograded** and feedback will be available shortly on the **Grades** tab.