P1

class Person {

    sayHello() {

        console.log("Hello");

    }

}

class Friend extends Person {

    sayHello() {

        console.log("Hey");

    }

}

var result = new Friend();

result.sayHello();

P2

class Animal {

}

class Cat extends Animal {

  constructor() {

    super();

    this.noise = "meow";

  }

}

var result = new Animal();

console.log(result.noise);

P3

class Animal {

    constructor(s) {

        this.s = s;

    }

}

class Dog extends Animal {

    constructor(lg,s) {

        super(s);

        this.lg=lg;

    }

}

var result=new Dog("hassan",2);

console.log(result.s,result.lg)

P4

// Task 1: Person class

class Person {

    constructor(name = "Tom", age = 20, energy = 100) {

      this.name = name;

      this.age = age;

      this.energy = energy;

    }

    sleep() {

      this.energy += 10;

    }

    doSomethingFun() {

      this.energy -= 10;

    }

  }

  // Task 2: Worker class (subclass of Person)

  class Worker extends Person {

    constructor(name = "Tom", age = 20, energy = 100, xp = 0, hourlyWage = 10) {

      super(name, age, energy);

      this.xp = xp;

      this.hourlyWage = hourlyWage;

    }

    goToWork() {

      this.xp += 10;

    }

  }

  // Task 3: Creating the intern object

  function createIntern() {

    const intern = new Worker("Bob", 21, 110, 0, 10);

    intern.goToWork();

    return intern;

  }

  // Task 4: Creating the manager object

  function createManager() {

    const manager = new Worker("Alice", 30, 120, 100, 30);

    manager.doSomethingFun();

    return manager;

  }

  // Test the code

  const internObject = createIntern();

  console.log("Intern Object:");

  console.log("Name:", internObject.name);

  console.log("Age:", internObject.age);

  console.log("Energy:", internObject.energy);

  console.log("XP:", internObject.xp);

  console.log("Hourly Wage:", internObject.hourlyWage);

  const managerObject = createManager();

  console.log("\nManager Object:");

  console.log("Name:", managerObject.name);

  console.log("Age:", managerObject.age);

  console.log("Energy:", managerObject.energy);

  console.log("XP:", managerObject.xp);

  console.log("Hourly Wage:", managerObject.hourlyWage);

  // Invoking the methods on a new instance of Person

  const PO = new Person();

  PO.sleep();

  console.log("Energy after sleep():", PO.energy);

  PO.doSomethingFun();

  console.log("Energy after doSomethingFun():", PO.energy);

**Additional resources**

Here is a list of resources that may be helpful as you continue your learning journey.

[Constructor](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes/constructor)

[Classes](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes)

[Object-oriented programming](https://css-tricks.com/the-flavors-of-object-oriented-programming-in-javascript/)

[Regular Expressions in JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions)

[RegExp object in JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp)