DAY 5

1. Write a “person” class to hold all the details.

class Person {

constructor(firstName, lastName, age, address) {

this.firstName = firstName;

this.lastName = lastName;

this.age = age;

this.address = address;

}

getFullName() {

return `${this.firstName} ${this.lastName}`;

}

getDetails() {

return `${this.getFullName()}, ${this.age} years old, from ${this.address}`;

}

updateAddress(newAddress) {

this.address = newAddress;

}

haveBirthday() {

this.age++;

}

}

// Example usage:

const johnDoe = new Person("John", "Doe", 30, "123 Main Street, Cityville");

console.log(johnDoe.getDetails()); // Output: John Doe, 30 years old, from 123 Main Street, Cityville

johnDoe.updateAddress("456 Oak Avenue, Townsville");

console.log(johnDoe.getDetails()); // Output: John Doe, 30 years old, from 456 Oak Avenue, Townsville

johnDoe.haveBirthday();

console.log(johnDoe.getDetails()); // Output: John Doe, 31 years old, from 456 Oak Avenue, Townsville

1. write a class to calculate the uber price.

class UberPriceCalculator {

constructor(baseFare, costPerKilometer, costPerMinute) {

this.baseFare = baseFare;

this.costPerKilometer = costPerKilometer;

this.costPerMinute = costPerMinute;

}

calculatePrice(distance, duration) {

const distanceCost = distance \* this.costPerKilometer;

const durationCost = duration \* this.costPerMinute;

const totalCost = this.baseFare + distanceCost + durationCost;

return totalCost;

}

}

// Example usage:

const uberXCalculator = new UberPriceCalculator(5, 1.5, 0.2);

const distance = 10; // in kilometers

const duration = 20; // in minutes

const totalPrice = uberXCalculator.calculatePrice(distance, duration);

console.log(`Estimated UberX Price: $${totalPrice.toFixed(2)}`);