1. **Personal Class with all details**

class Person {

constructor(name, email, phone, website, address, degree, cgp, college, university, experiences) {

this.name = name;

this.email = email;

this.phone = phone;

this.website = website;

this.address = address;

this.degree = degree;

this.cgp = cgp;

this.college = college;

this.university = university;

this.experiences = experiences;

}

getPersonalInfo() {

return {

name: this.name,

email: this.email,

phone: this.phone,

website: this.website,

address: this.address

};

}

getEducationInfo() {

return {

degree: this.degree,

cgp: this.cgp,

college: this.college,

university: this.university

};

}

getExperienceInfo() {

return this.experiences;

}

addExperience(role, company, duration) {

this.experiences.push({ role, company, duration });

}

getAllDetails() {

return {

personal\_info: this.getPersonalInfo(),

education: this.getEducationInfo(),

experience: this.getExperienceInfo()

};

}

}

const experiences = [

{ role: "Full Stack Developer", company: "Bug Hunt Technologies", duration: "1 year" },

{ role: "Hardware Technician", company: "ABC Tech", duration: "6 months" },

{ role: "UI/UX Designer", company: "Creative Designs", duration: "6 months" }

];

const person = new Person(

"Vigneshwaran J",

"venerablevignesh@gmail.com",

"6380661438",

"mastercodeaddict.me",

"697/1, Thirupathi venkateshwara nagar, Manaveli, Thiruvanthipuram, Cuddalore",

"Bachelor of Science in Information Technology",

7.9,

"Achariya Arts and Science college",

"Pondicherry University",

experiences

);

console.log("Personal Details");

person.addExperience("Project Manager", "XYZ Corp", "2 years");

console.log(person.getAllDetails());

console.log("--------------------------------------");

1. **UBER RIDE CALCULAOTORE**

class UberRide {

constructor(baseFare, costPerMinute, costPerMile, surgeMultiplier = 1, bookingFee = 0, minimumFare = 0) {

this.baseFare = baseFare;

this.costPerMinute = costPerMinute;

this.costPerMile = costPerMile;

this.surgeMultiplier = surgeMultiplier;

this.bookingFee = bookingFee;

this.minimumFare = minimumFare;

}

calculatePrice(distance, duration) {

const distanceCost = this.costPerMile \* distance;

const timeCost = this.costPerMinute \* duration;

const surgeCost = (this.baseFare + distanceCost + timeCost) \* this.surgeMultiplier;

const totalPrice = surgeCost + this.bookingFee;

return Math.max(totalPrice, this.minimumFare);

}

}

const uberRide = new UberRide(

2.50, 0.35, 1.75, 1.2, 2.0, 5.0

);

const distance = 10;

const duration = 15;

console.log("Uber Ride Price calculatore");

const price = uberRide.calculatePrice(distance, duration);

console.log(`Total Uber ride price: $${price.toFixed(2)}`);

console.log("------------------------------");