**1. Class - Movie**

**The class Movie is stated below. An instance of class Movie represents a film. This class has the following three properties:**

**title, which is a String representing the title of the movie**

**studio, which is a String representing the studio that made the movie**

**rating, which is a String representing the rating of the movie (i.e. PG­13, R, etc)**

**a) Write a constructor for the class Movie, which takes a String representing the title of the movie, a String representing the studio, and a String representing the rating as its arguments, and sets the respective class properties to these values.**

**b) The constructor for the class Movie will set the class property rating to "PG" as default when no rating is provided.**

**c) Write a method getPG, which takes an array of base type Movie as its argument, and returns a new array of only those movies in the input array with a rating of "PG". You may assume the input array is full of Movie instances. The returned array need not be full.**

**d) Write a piece of code that creates an instance of the class Movie with the title “Casino Royale”, the studio “Eon Productions”, and the rating “PG­13”**

**a) Constructor for the class Movie:**

public class Movie {

private String title;

private String studio;

private String rating;

public Movie(String title, String studio, String rating) {

this.title = title;

this.studio = studio;

if (rating != null) {

this.rating = rating;

} else {

this.rating = "PG";

}

}

// getters and setters omitted for brevity

}

**b) The constructor sets the rating property to "PG" by default if no rating is provided.**

**c) Method getPG:**

public Movie[] getPG(Movie[] movies) {

List<Movie> pgMovies = new ArrayList<>();

for (Movie movie : movies) {

if ("PG".equals(movie.getRating())) {

pgMovies.add(movie);

}

}

return pgMovies.toArray(new Movie[0]);

}

**d) Code to create an instance of the class Movie:**

Movie casinoRoyale = new Movie("Casino Royale", "Eon Productions", "PG-13");

**2. Circle - Class**

**Convert the UML diagram to Typescript class. - use number for double**

class Circle {

private radius: number;

private color: string;

constructor(radius: number, color: string) {

this.radius = radius;

this.color = color;

}

getRadius(): number {

return this.radius;

}

setRadius(radius: number): void {

this.radius = radius;

}

getColor(): string {

return this.color;

}

setColor(color: string): void {

this.color = color;

}

getArea(): number {

return Math.PI \* Math.pow(this.radius, 2);

}

getCircumference(): number {

return 2 \* Math.PI \* this.radius;

}

}

This Circle class has private properties for the radius and color, as well as getter and setter methods for each property. Additionally, it includes methods for calculating the area and circumference of the circle.

You can create a new instance of the Circle class and use its methods:

let circle = new Circle(5, 'red');

console.log(circle.getRadius()); // Output: 5

console.log(circle.getColor()); // Output: "red"

console.log(circle.getArea()); // Output: 78.53981633974483

console.log(circle.getCircumference()); // Output: 31.41592653589793

circle.setRadius(10);

circle.setColor('blue');

console.log(circle.getRadius()); // Output: 10

console.log(circle.getColor()); // Output: "blue"

console.log(circle.getArea()); // Output: 314.1592653589793

console.log(circle.getCircumference()); // Output: 62.83185307179586

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

class Person {

private firstName: string;

private lastName: string;

private age: number;

private address: string;

private email: string;

private phone: string;

constructor(firstName: string, lastName: string, age: number, address: string, email: string, phone: string) {

this.firstName = firstName;

this.lastName = lastName;

this.age = age;

this.address = address;

this.email = email;

this.phone = phone;

}

getFirstName(): string {

return this.firstName;

}

setFirstName(firstName: string): void {

this.firstName = firstName;

}

getLastName(): string {

return this.lastName;

}

setLastName(lastName: string): void {

this.lastName = lastName;

}

getAge(): number {

return this.age;

}

setAge(age: number): void {

this.age = age;

}

getAddress(): string {

return this.address;

}

setAddress(address: string): void {

this.address = address;

}

getEmail(): string {

return this.email;

}

setEmail(email: string): void {

this.email = email;

}

getPhone(): string {

return this.phone;

}

setPhone(phone: string): void {

this.phone = phone;

}

}

This Person class has private properties for first name, last name, age, address, email, and phone, as well as public getter and setter methods for each property. You can create a new instance of the Person class and set its properties:

let person = new Person("John", "Doe", 30, "123 Main St", "john.doe@example.com", "555-555-5555");

person.setAge(31);

console.log(person.getAge()); // Output: 31

**4. write a class to calculate the uber price.**

class UberPriceCalculator {

private baseFare: number = 2.5;

private ratePerMile: number = 1.5;

private ratePerMinute: number = 0.25;

calculatePrice(distance: number, time: number): number {

const distanceInMiles = distance / 1609.34; // Convert meters to miles

const pricePerMile = this.ratePerMile \* distanceInMiles;

const pricePerMinute = this.ratePerMinute \* time;

const totalPrice = this.baseFare + pricePerMile + pricePerMinute;

return totalPrice;

}

}

This UberPriceCalculator class has private properties for the base fare, rate per mile, and rate per minute, as well as a calculatePrice method that takes the distance (in meters) and time (in seconds) as arguments and returns the calculated price. The distance parameter is converted to miles by dividing by 1609.34.

You can create a new instance of the UberPriceCalculator class and use the calculatePrice method to calculate the price of an Uber ride:

let calculator = new UberPriceCalculator();

let distance = 5000; // 5 kilometers

let time = 600; // 10 minutes

let price = calculator.calculatePrice(distance, time);

console.log(price); // Output: 11.75