**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”**

class Movie {

constructor(title, studio, rating = "PG") {

this.title = title;

this.studio = studio;

this.rating = rating;

}

static getPG(movies) {

return movies.filter(movie => movie.rating === "PG");

}

}

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

console.log("Movie Title:", casinoRoyale.title);

console.log("Studio:", casinoRoyale.studio);

console.log("Rating:", casinoRoyale.rating);

const moviesArray = [

new Movie("Movie1", "Studio1", "PG"),

new Movie("Movie2", "Studio2", "R"),

new Movie("Movie3", "Studio3", "PG-13"),

new Movie("Movie4", "Studio4"),

];

const pgMovies = Movie.getPG(moviesArray);

console.log("\nPG Rated Movies:");

pgMovies.forEach(movie => {

console.log("Movie Title:", movie.title);

console.log("Studio:", movie.studio);

console.log("Rating:", movie.rating);

console.log("---");

});

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

class Person {

constructor(firstName, lastName, age, gender, email) {

this.firstName = firstName;

this.lastName = lastName;

this.age = age;

this.gender = gender;

this.email = email;

}

getFullName() {

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

}

getAge() {

return this.age;

}

getGender() {

return this.gender;

}

getEmail() {

return this.email;

}

}

// Example usage:

const person1 = new Person("John", "Doe", 30, "Male", "john.doe@example.com");

console.log("Full Name:", person1.getFullName());

console.log("Age:", person1.getAge());

console.log("Gender:", person1.getGender());

console.log("Email:", person1.getEmail());

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

class UberPriceCalculator {

constructor(distance, duration, rideType) {

this.distance = distance;

this.duration = duration;

this.rideType = rideType;

}

calculatePrice() {

const baseFare = 5;

const costPerKilometer = 1.5;

const costPerMinute = 0.2;

const additionalCosts = {

"UberX": 0,

"UberXL": 3,

};

const totalFare =

baseFare +

costPerKilometer \* this.distance +

costPerMinute \* this.duration +

additionalCosts[this.rideType];

return totalFare;

}

}

const uberRide = new UberPriceCalculator(10, 20, "UberX");

const price = uberRide.calculatePrice();

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