1Q)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”

**Solution:**

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

java

public class Movie {

private String title;

private String studio;

private String rating;

// Constructor with title, studio, and rating arguments

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

this.title = title;

this.studio = studio;

// If no rating is provided, set rating to "PG" as default

if (rating == null || rating.isEmpty()) {

this.rating = "PG";

} else {

this.rating = rating;

}

}

}

**b) Default rating:**

The constructor for the class Movie will set the class property rating to "PG" as default when no rating is provided. This is achieved in the above constructor by checking if the provided rating is null or empty, and setting it to "PG" in that case.

**c) Method getPG:**

java

// Method to get PG-rated movies from an array of Movie instances

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

int countPG = 0;

// Count the number of PG-rated movies

for (Movie movie : movies) {

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

countPG++;

}

}

// Create a new array for PG-rated movies

Movie[] pgMovies = new Movie[countPG];

int index = 0;

// Populate the new array with PG-rated movies

for (Movie movie : movies) {

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

pgMovies[index] = movie;

index++;

}

}

return pgMovies;

}

**d) Creating an instance of the class Movie:**

java

// Creating an instance of Movie class with the title “Casino Royale”, the studio “Eon Productions”, and the rating “PG­13”

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

**The complete code is:**

public class Movie {

private String title;

private String studio;

private String rating;

// Constructor with title, studio, and rating arguments

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

this.title = title;

this.studio = studio;

// If no rating is provided, set rating to "PG" as default

if (rating == null || rating.isEmpty()) {

this.rating = "PG";

} else {

this.rating = rating;

}

}

// Getter for the rating property

public String getRating() {

return rating;

}

// Method to get PG-rated movies from an array of Movie instances

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

int countPG = 0;

// Count the number of PG-rated movies

for (Movie movie : movies) {

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

countPG++;

}

}

// Create a new array for PG-rated movies

Movie[] pgMovies = new Movie[countPG];

int index = 0;

// Populate the new array with PG-rated movies

for (Movie movie : movies) {

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

pgMovies[index] = movie;

index++;

}

}

return pgMovies;

}

public static void main(String[] args) {

// Creating an instance of Movie class with provided details

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

}

}

3Q)

class Person {

private name: string;

private age: number;

private gender: string;

constructor(name: string, age: number, gender: string) {

this.name = name;

this.age = age;

this.gender = gender;

}

getName(): string {

return this.name;

}

setName(name: string): void {

this.name = name;

}

getAge(): number {

return this.age;

}

setAge(age: number): void {

this.age = age;

}

getGender(): string {

return this.gender;

}

setGender(gender: string): void {

this.gender = gender;

}

}

class UberPriceCalculator {

private baseFare: number;

private perDistanceRate: number;

private perMinuteRate: number;

constructor(baseFare: number, perDistanceRate: number, perMinuteRate: number) {

this.baseFare = baseFare;

this.perDistanceRate = perDistanceRate;

this.perMinuteRate = perMinuteRate;

}

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

// Assuming distance is in kilometers and duration is in minutes

const price = this.baseFare + (this.perDistanceRate \* distance) + (this.perMinuteRate \* duration);

return price;

}

}

// Example usage:

const person = new Person("John", 30, "male");

const uberPriceCalculator = new UberPriceCalculator(5, 1.5, 0.5);

const distance = 10; // in kilometers

const duration = 20; // in minutes

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

console.log(`Uber price for ${person.getName()}: $${price}`);