Task 6

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;

    }

    getPG(movies){

        let pgRatingMovies = [];

        for (let movie of movies){

            if (movie.rating === "PG"){

                pgRatingMovies.push(movie);

            }

        }

        return pgRatingMovies;

    }

}

casinoRoyale = new Movie("Casino Royale", "Eon Productions", "PG13");

movie1 = new Movie("abc","124");

movie2 = new Movie("aaa","123","PG1");

movie3 = new Movie("bbb","234","PG12");

movie4 = new Movie("ccc","345","PG");

movie5 = new Movie("ddd","567","PG");

movielist = [movie1,movie2,movie3,movie4,movie5]

console.log(casinoRoyale.getPG(movielist))

2)

Create a class circle

class Circle{

    constructor(radius,color){

        this.radius = radius;

        this.color = color;

    }

    getRadius(){

        return this.radius;

    }

    setRadius(radius){

        this.radius = radius;

    }

    getColor(){

        return this.color;

    }

    setColor(color){

        this.color = color;

    }

    toString(){

        return `Circle -Radius ${this.radius}, Color ${this.color}`

    }

    getArea(){

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

    }

    getCircumference(){

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

    }

}

sample = new Circle(2,"Pink");

console.log(sample.getRadius());

console.log(sample.toString())

console.log(sample.getCircumference());

3) Write a person class to hold all details

class Person {

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

      this.firstName = firstName;

      this.lastName = lastName;

      this.age = age;

      this.email = email;

      this.gender = gender;

      this.friends = [];

    }

    setFirstName(firstName){

        this.firstName = this.firstName;

    }

    getFirstName(){

        return this.firstName;

    }

    setLastName(lastName){

        this.lastName = lastName;

    }

    getLastName(){

        return this.lastName();

    }

    getFullName(){

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

    }

    setGender(gender){

        this.setGender(gender);

    }

    getGender(){

        return this.gender;

    }

    setAge(age){

        this.setAge(age);

    }

    getAge(){

        return this.getAge;

    }

    setEmail(email){

        this.email = email;

    }

    getEmail(){

        return this.email;

    }

    addFriend(friend){

        this.friends.push(friend);

    }

    getFriends(){

        return this.friends;

    }

}

person1  =  new Person("John","Baby",30,"johnbaby@gmail.com","male");

person1.addFriend("Abraham");

person1.addFriend("Julie");

person1.addFriend("Cathy");

console.log(person1.getFullName)

for(let item of person1.friends){

    console.log(item);

}

4) Write a class to calculate the Uber Price

class CabPriceCalculator {

    constructor(baseFare, costPerKm) {

      this.baseFare = baseFare;

      this.costPerKm = costPerKm;

    }

    calculatePrice(distance) {

      const distanceCost = distance \* this.costPerKm;

      const totalPrice = this.baseFare + distanceCost;

      return totalPrice;

    }

  }

  // Example usage:

  const calculator = new CabPriceCalculator(2.5, 50); // Example rates

  const distance = 10; // in kilometers

  const totalPrice = calculator.calculatePrice(distance);

  console.log(`Total price for the ride: Rs ${totalPrice.toFixed(2)}`);

O/p : Total price for the ride: Rs 502.50