

1. <https://github.com/rvsp/typescript-oops/blob/master/Practice/Movie.md>

```
class Movie {
  constructor(title, studio, rating = "PG") {
    this.title = title;
    this.studio = studio;
    this.rating = rating;
  }
  static getPG(arr) {
    let pgArray = arr.filter((movie) => movie.rating == "PG");
    return pgArray;
  }
}

let a = new Movie("Casino Royale", "Eon Productions", "PG13");
let b = new Movie("Inception", "Warner Bros", "PG");
let c = new Movie("Iron Man", "Marvel Studios");
let d = new Movie("Orphan", "Dark Castle Entertainment", "PG13");
let e = new Movie("Cars", "Pixar", "G");

console.log(Movie.getPG([a, b, c, d, e]));
```

2. <https://github.com/rvsp/typescript-oops/blob/master/Practice/class-circle.md>

```
class Circle {
  constructor(...arr) {
    this.radius = arr[0];
    this.color = arr[1];

    this.arr = arr.filter((i) => i != undefined);

    if (this.arr.length == 0) {
      console.log("Empty")
    } else if (this.arr.length == 1) {
      console.log(`Radius : ${this.radius}`)
    } else {
      console.log(`Radius : ${this.radius}, Color : ${this.color}`)
    }
  }

  getRadius() {
    return this.radius;
  }

  getColor() {
    return this.color;
  }
}
```

```

    setRadius(rad) {
        this.radius = rad;
    }

    setColor(col) {
        this.color = col;
    }

    toString(){
        return `Radius : ${this.radius}, Color : ${this.color}`;
    }

    getArea(){
        return Math.PI * Math.pow(this.radius, this.radius);
    }
}

let a = new Circle()
let b = new Circle(5.5)
let c = new Circle(5.5, "red")

```

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

```

class person {
    constructor (firstName, lastName, age) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.age = age;
        this.getFullName = function () {
            return this.firstName + " " + this.lastName;
        }
    }
}

let person1 = new person ("kumar", "ramu", 35 );
console.log(person1.age);
console.log(person1.getFullName());

```

4. write a class to calculate the uber price.

```

class uberFare {
    constructor (city, fare, customerName) {
        this.city = city;
        this.fare = fare;
        this.customerName = customerName;
    }
}

```

```
    }  
    getFareCity(){  
        return "fare for the City is " + this.fare;  
    }  
}  
  
let cityA = new uberFare( "cityA", 600, "customer1");  
let cityB = new uberFare( "cityB", 300, "customer2");  
  
console.log(cityA.getFareCity());  
console.log(cityB.getFareCity());
```