DAY 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(){
        console.log(this.title,this.studio,this.rating);
    }
```

```
}
let mov=new Movie('Casino Royale','Eon Productions','PG13');

let mov1=new Movie('Casino Royale','Eon Productions');

mov.getPG();
mov1.getPG();
```

Output:

```
Casino Royale Eon Productions PG13
Casino Royale Eon Productions PG
```

```
//2.circle of properties in Class


class Circle{
    constructor(radius, color){
        this.radius = radius;
        this.color = color;
    }
    get radiusCircle(){
        return this.radius;
    }
    set radiusCircle(radius){
            this.radius = radius;
    }
    get colorCircle(){
        return this.color;
    }
    set colorCircle(color){
        this.color = color;
    }
    get toString(){
        return (`"Circle[radius=${this.radius},colors=${this.color}]"`);
    }
    get areaCircle(){
        return Math.PI * this.radius * this.radius;
    }
    get circumferenceCircle(){
        return 2 * Math.PI * this.radius;
    }
}
var obj1 = new Circle(5.0,"Orange");

console.log(obj1.radiusCircle);

obj1.colorCircle = "blue";

console.log(obj1.colorCircle);

console.log(obj1.toString);

console.log("Area of Circle:",obj1.areaCircle);

console.log("Circumference of Circle:",obj1.circumferenceCircle);
```

Output:

```
5
blue
"Circle[radius=5,colors=blue]"
Area of Circle: 78.53981633974483
Circumference of Circle: 31.41592653589793
```

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

class Person{
    constructor(){
        this.name='noname';
        this.age=0;
        this.city='none';
        this.email='no@gmail.com';
    }
    getDetails(){
        console.log(`Name:
${this.name},\n`,`Age:${this.age},\n`,`City:${this.city},\n`,`Email:${this
.email}`);
    }
}

let person1=new Person();
let person2=new Person();

person1.name='Prakash';
person1.age='22';
person1.city='trichy';
person1.email='prakash123@gmail.com'

person2.name='Vijay';
person2.age='22';
person2.city='Madurai';
person2.email='vijay123@gmail.com'


person1.getDetails();
person2.getDetails();
```

Output:

```
Name: Prakash,
 Age:22,
 City:trichy,
 Email:prakash123@gmail.com
Name: Vijay,
 Age:22,
 City:Madurai,
 Email:vijay123@gmail.com
```

```
//4.write a class to calculate the Uber price.

class UberPricing {
    constructor(baseFare, costPerKm, costPerMinute) {
      this.baseFare = baseFare;
      this.costPerKm = costPerKm;
      this.costPerMinute = costPerMinute;
    }

    calculatePrice(distanceKm, timeMinutes) {
      const fare = this.baseFare + (distanceKm * this.costPerKm) +
(timeMinutes * this.costPerMinute);
      return fare;
    }
  }

  const baseFare=3;
  const constperKM=10;
  const costPerMinute=1.5;

  const uberX = new UberPricing(baseFare,constperKM,costPerMinute);

  const distance = 10;
  const time = 20;

  const price = uberX.calculatePrice(distance, time);
  console.log("Estimated UberX Price:", price);
```

Output:

```
Estimated UberX Price: 133
```