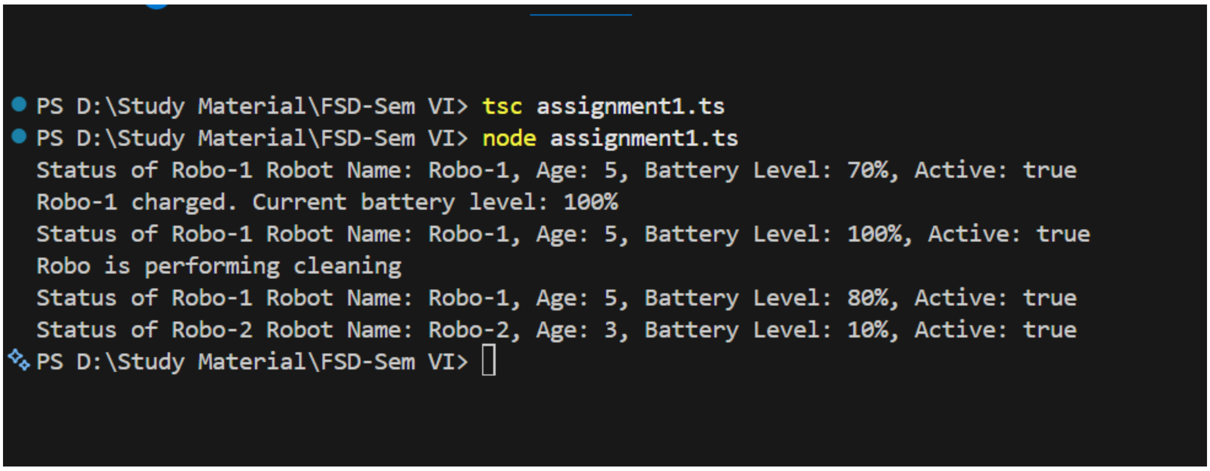
Type Script Assignment:

- **Code:**

```
interface Robot {
    name: string;
    age: number;
    batteryLevel: number;
    isActive: boolean;
    chargeBattery(amount: number): void;
    performTask(task: string): string;
}

class RobotClass implements Robot {
    name: string;
    age: number;
    batteryLevel: number;
    isActive: boolean;
    constructor(name: string, age: number, batteryLevel: number) {
        this.name = name;
        this.age = age;
        this.batteryLevel = batteryLevel;
        this.isActive = true;
    }
    chargeBattery(amount: number): void {
        this.batteryLevel += amount;
        if (this.batteryLevel > 100) {
            this.batteryLevel = 100;
        }
        console.log(`${this.name} charged. Current battery level: ${this.batteryLevel}%`);
    }
    performTask(task: string): string {
        if (!this.isActive) {
            return "Robot is inactive.";
        }
        if (this.batteryLevel <= 20) {
            return "Battery is too low to perform tasks!";
        }
        this.batteryLevel -= 20;
        return `Robo is performing ${task}`;
    }

    status = (): string => {
        return `Robot Name: ${this.name}, Age: ${this.age}, Battery Level: ${this.batteryLevel}%, Active: ${this.isActive}`;
    }
    shutdown = (): void => {
        this.isActive = false;
        console.log(`${this.name} is shutting down.`);
    }
}
class RobotManager{
    robots: RobotClass[] = [];

    addRobot(robot: RobotClass): void {
        this.robots.push(robot);
    }
    listRobots(): void {
        this.robots.forEach(robot => {
            console.log(robot.status());
        });
    findRobotByName(name: string): RobotClass | undefined {
        return this.robots.find(robot => robot.name === name);
    }
    removeRobotByName(name: string): void {
        this.robots = this.robots.filter(robot => robot.name !== name);
    }

    }
```

```
const Robot1 = new RobotClass('Robo-1',5,70);
const Robot2 = new RobotClass('Robo-2',3,10);


console.log('Status of Robo-1',Robot1.status());
Robot1.chargeBattery(30);
console.log('Status of Robo-1',Robot1.status());
console.log(Robot1.performTask("cleaning"));
console.log('Status of Robo-1',Robot1.status());


console.log('Status of Robo-2',Robot2.status());
```

- **Output:**



```
PS D:\Study Material\FSD-Sem VI> tsc assignment1.ts
PS D:\Study Material\FSD-Sem VI> node assignment1.ts
Status of Robo-1 Robot Name: Robo-1, Age: 5, Battery Level: 70%, Active: true
Robo-1 charged. Current battery level: 100%
Status of Robo-1 Robot Name: Robo-1, Age: 5, Battery Level: 100%, Active: true
Robo is performing cleaning
Status of Robo-1 Robot Name: Robo-1, Age: 5, Battery Level: 80%, Active: true
Status of Robo-2 Robot Name: Robo-2, Age: 3, Battery Level: 10%, Active: true
PS D:\Study Material\FSD-Sem VI> 
```