Object Oriented Programming

Class

> Class is a template or model for a real world object.
> Class has some data and behaviors.
> The data and behaviors are called attributes and methods.

Class structure

```
class Student {

    constructor() {

    }
}
```

Class with data

```
class Student {

    constructor(name, year, degree) {
        this.name = name;
        this.year = year;
        this.degree = degree;

    }
}
// All the above properties are public. Others can see.
```

Create Object from Class

```
var meghaa = new Student('Meghaa',"Second", "M.C.A");
var suganthi = new Student('Suganthi', "Third", "B.E");
console.log(meghaa); // Student { name: 'Meghaa', year: 'Second', degree: 'M.C.A'
}
```

Class with private fields

```
class Student {
    #age = 0; // this is a private field
    constructor(name, year, degree, age) {
        this.name = name; // public field
        this.year = year;
        this.degree = degree;
        this.#age = age;


    }
}

var meghaa = new Student('Meghaa',"Second", "M.C.A", 20);
console.log(meghaa); // Student { name: 'Meghaa', year: 'Second', degree: 'M.C.A'
}
// The age is not printed out.
```

Methods in a class

```
class Student {
    #age = 0; // this is a private field
    constructor(name, year, degree, age) {
        this.name = name; // public field
        this.year = year;
        this.degree = degree;
        this.#age = age;

    }

  getStudentDetails(){
      return `${this.name}, ${this.#age} `;
  }

   updateAge(age){
    this.#age = age;
  }
}
```

Encapsulation : restricting access to data.

Make all data private and have access through only methods.

```
class Student {
    #age = 0; // this is a private field
    #name;
    #year;
    #degree
    constructor(name, year, degree, age) {
        this.#name = name; // public field
        this.#year = year;
        this.#degree = degree;
        this.#age = age;

    }

    getStudentDetails(){
        return `${this.#name}, ${this.#age} `;
    }

    updateAge(age){
      this.#age = age;
    }
}

console.log(meghaa); // Student {}
meghaa.updateAge(10);
console.log(meghaa.getStudentDetails()); // Meghaa, 10
```