

# JAVA OOP

---

CLASS AND OBJECT

## 2 INTRODUCTION

---

- Java គឺជា object-oriented programming language ។ គំនិតស្នូលនៃវិធីសាស្ត្រ object-oriented គឺបំបែកបញ្ហាស្មុគស្មាញទៅជា Object តូចៗ។ Object គឺជា entity ដែលមាន state និង behavior ។
- State ប្រាប់អំពីប្រភេទ លក្ខណៈ ទំរង់ សណ្ឋាន តំលៃនៃ object ។
- Behavior ប្រាប់ពីប្រតិបត្តិការ ឬ សកម្មភាពដែល object អាចសម្តែងឡើង។
- ឧ. ឡាន គឺជា object ។ វាមាន state ដូចជា ពណ៌ ប្រភេទម៉ាស៊ីន កង់ ។ល។ ឡានអាចបើកក្នុងល្បឿន ១៨០km/h វាអាចបត់ធ្វេងស្តាំ ទៅមុខថយក្រោយ អាចផ្ទុកមនុស្ស ទាំងនេះជា behavior របស់ឡាន។

## 3 CLASS

---

- Class គឺជាពុម្ពគំរូ (Blueprint) របស់ object ។ មុននឹងយើងបង្កើត object យើងត្រូវកំណត់ Class ជាមុនសិន។
- ការបង្កើត Class

```
class ClassName {  
  
    // fields  
  
    // methods  
  
}
```

## 4 CLASS

---

- Field (variable) និង Method តំណាងអោយ state និង behavior
- Field ប្រើសម្រាប់ផ្ទុកទិន្នន័យ
- Method ប្រើដើម្បីសម្តែងប្រតិបត្តិការ
- Field និង Method គឺជា member នៃ class ។

## 5 CLASS

---

```
class Car{  
    String color; // state or field  
    public void drive(){ // behavior or method  
        System.out.print("car drive");  
    }  
}
```



## 6 OBJECT

---

- Object គឺជាផលសម្រេច (instance) ដែលកើតចេញពី class។ ឧបមាថា Car គឺជា class មួយ នោះ sportCar, touringCar អាចចាត់ថ្នាក់ជា object របស់ class Car ។
- ការបង្កើត Object

```
className object = new className();
```

```
Car sportCar = new Car();
```

```
Car touringCar = new Car();
```

## 7 ACCESS MEMBER OF CLASS

---

យើងអាចប្រើឈ្មោះរបស់ object ភ្ជាប់ជាមួយសញ្ញា . ដើម្បី access member របស់ class ។

```
sportCar.color;
```

```
sportCar.drive();
```

## 8 CLASS AND OBJECT

---

```
Class Lamp{ // អំពូល
    Boolean isOn;
    void turnOn() {
        isOn = true;
        System.out.println("Light on? " + isOn);
    }
}
```



## 9 CLASS AND OBJECT

---

```
Class Main{
```

```
    public static void main(String[] args) {
```

```
        Lamp smallLamp = new Lamp();
```

```
        smallLamp.turnOn();
```

```
    }
```

```
}
```

Output: Light on? true