Create a class in Java

We can create a class in Java using the class keyword. For example,

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

Here, `fields` (variables) and `methods` represent the state and behavior of the object respectively.

- fields are used to store data
- methods are used to perform some operations

Java Objects

An object is called an instance of a class. For example, suppose **`Bicycle`** is a class then **`MountainBicycle`**, **`SportsBicycle`**, **`TouringBicycle`**, etc can be considered as objects of the class.

Creating an Object in Java

Here is how we can create an object of a class.

```
className object = new className();

// for Bicycle class
Bicycle sportsBicycle = new Bicycle();

Bicycle touringBicycle = new Bicycle();
```

We have used the `new` keyword along with the constructor of the class to create an object. Constructors are similar to methods and have the same name as the class. For example, `Bicycle()` is the constructor of the `Bicycle` class.

Access Members of a Class

We can use the name of objects along with the `.` operator to access members of a class. For example,

```
class Bicycle {

  // field of class
  int gear = 5;

  // method of class
  void braking() {
    ...
  }
}

// create object
Bicycle sportsBicycle = new Bicycle();
```

```
14
15      // access field and method
16      sportsBicycle.gear;
17      sportsBicycle.braking();
```

In the above example, we have created a class named `Bicycle`. It includes a field named `gear` and a method named `braking()`. Notice the statement,

Bicycle sportsBicycle = new Bicycle();

Here, we have created an object of `Bicycle` named `sportsBicycle`. We then use the object to access the field and method of the class.

- sportsBicycle.gear - access the field `gear`
- sportsBicycle.braking() - access the method `braking()`

Example: Java Class and Objects

```java
class Lamp {

  // stores the value for light
  // true if light is on
  // false if light is off
  boolean isOn;

  // method to turn on the light
  void turnOn() {
    isOn = true;
    System.out.println("Light on? " + isOn);

  }

  // method to turnoff the light
  void turnOff() {
    isOn = false;
    System.out.println("Light on? " + isOn);
  }
}

class Main {
  public static void main(String[] args) {

    // create objects led and halogen
    Lamp led = new Lamp();
    Lamp halogen = new Lamp();

    // turn on the light by
    // calling method turnOn()
    led.turnOn();

    // turn off the light by
    // calling method turnOff()
    halogen.turnOff();
  }
}
```

Output:

Light on? true
Light on? false

- led.turnOn() - It sets the `isOn` variable to `true` and prints the output.
- halogen.turnOff() - It sets the `isOn` variable to `false` and prints the output.