CLASS & OBJECTS

Control Statements

Recap

Let's Start !

Recap

Which **loop** will always execute at least once

✓ Do-While Loop

Recap

What is the purpose of the "break" statement in a loop?

✓ **Terminates the loop and transfers control to the next statement after the loop**

Recap

What is the result of the expression

(true && false) || (true || false)?

- ✓ **true**

OBJECTIVES

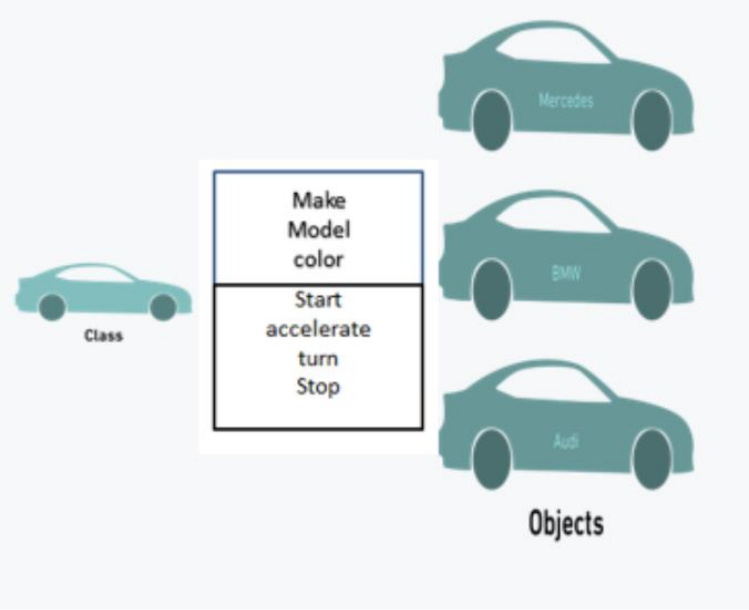
On completion of this topic, you will be able to:

- Explain Class and Object ✓
- Create Class and Object ✓
- Use appropriate Access Specifier ✓
- Write Getters, Setters Method ✓

CLASS & OBJECTS

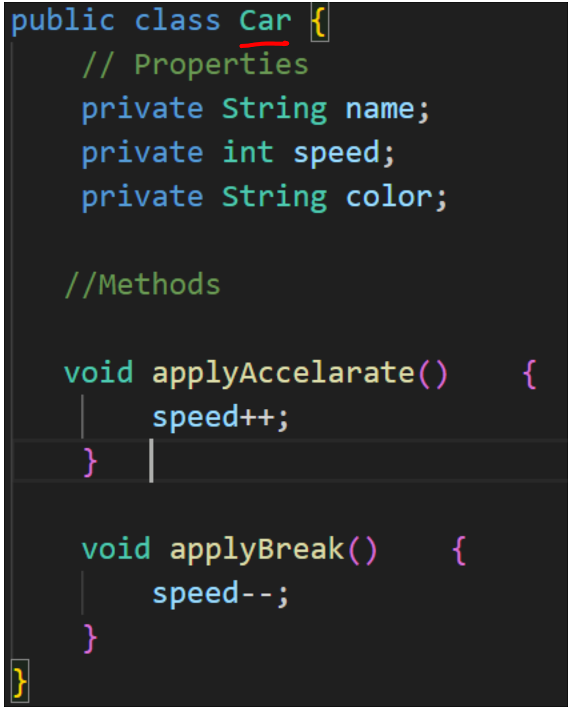
- **What is an Object**
  - Any real-world entity with a well-defined property is called an object
  - Object can be tangible (physical entity) or Intangible (cannot be touched or felt)

CLASS & OBJECTS



- **What is Class?**
  - A class is a template or a blueprint that describes the behaviours/state that an object of its type supports.
  - Object is an instance of a class

CLASS vs OBJECTS

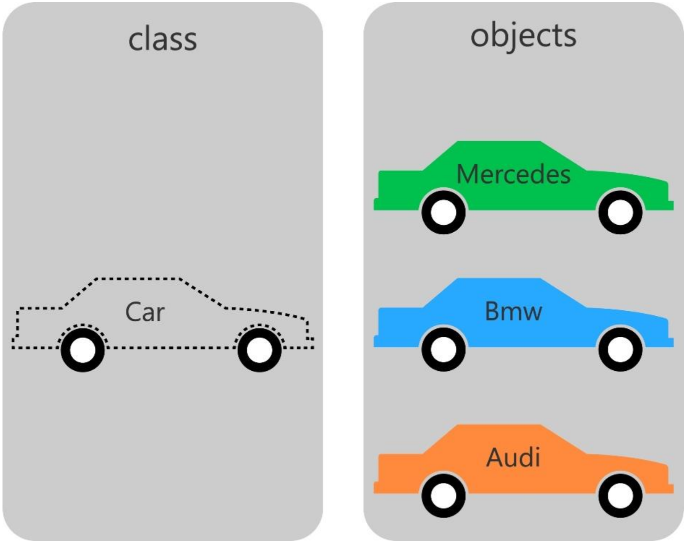


```
public class Car {
    // Properties
    private String name;
    private int speed;
    private String color;

   //Methods

   void applyAccelarate()     {
        speed++;
    }

    void applyBreak()    {
        speed--;
    }
}
```



name : Mercedes
color : Green
speed : 150

applyAccelarate(){...}
applyBreak() {…}

name : Bmw
color : Blue
speed : 160

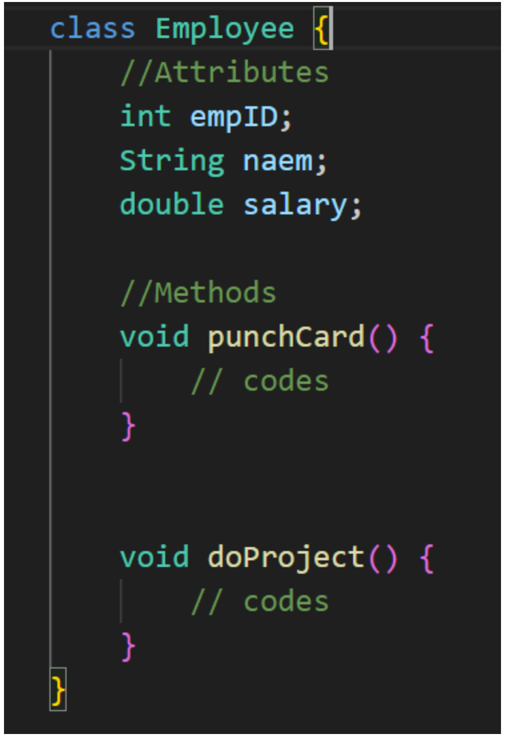
applyAccelarate(){...}
applyBreak() {…}

name : Bmw
color : Blue
speed : 160

applyAccelarate(){...}
applyBreak() {…}

CLASS vs OBJECTS

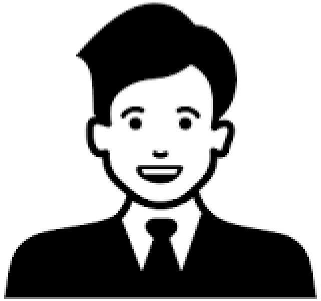
Another example



```
class Employee {
    //Attributes
    int empID;
    String naem;
    double salary;

    //Methods
    void punchCard() {
        // codes
    }


    void doProject() {
        // codes
    }
}
```



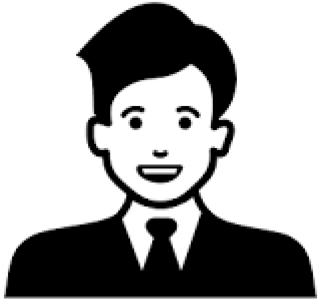
empID: 1001
name : Akil
salary : 50000.0

Object

CLASS vs OBJECTS

PROPERTIES OF AN OBJECT

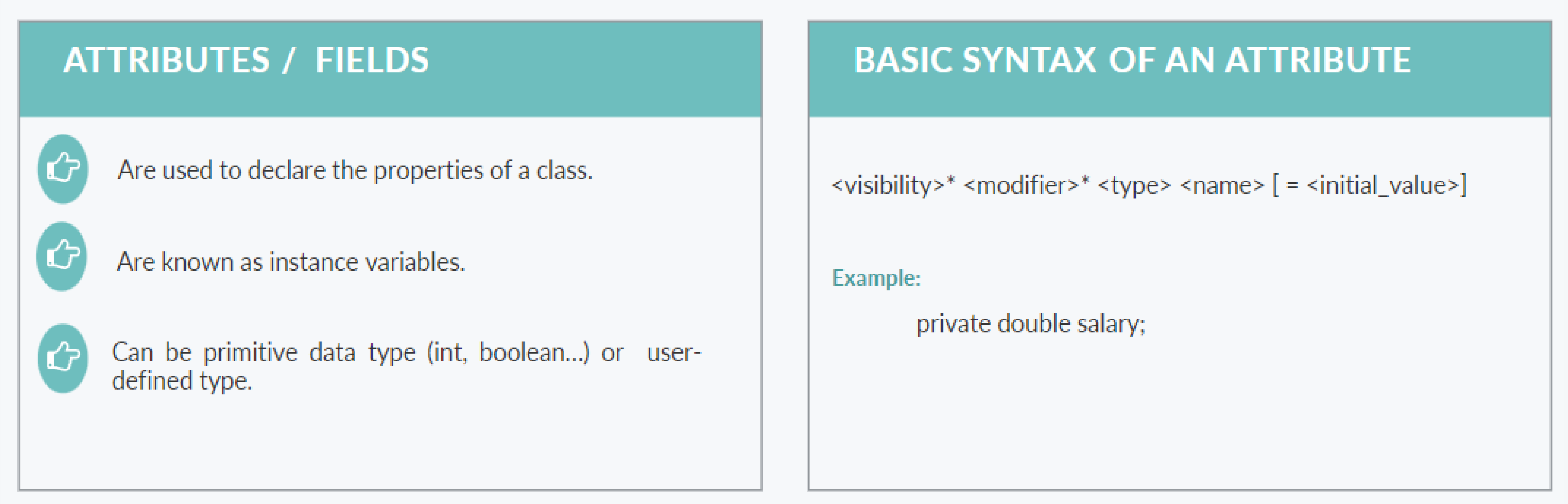
- **State :** Attribute + Value.
- **Behavior :** Operation that the object can perform.
- **Identify :** Every object should have a unique identity.



empID: 1001
name : Akil
salary : 50000.0

Object

ATTRIBUTE DECLARATION



ATTRIBUTES / FIELDS

- Are used to declare the properties of a class.
- Are known as instance variables.
- Can be primitive data type (int, boolean…) or user-defined type.

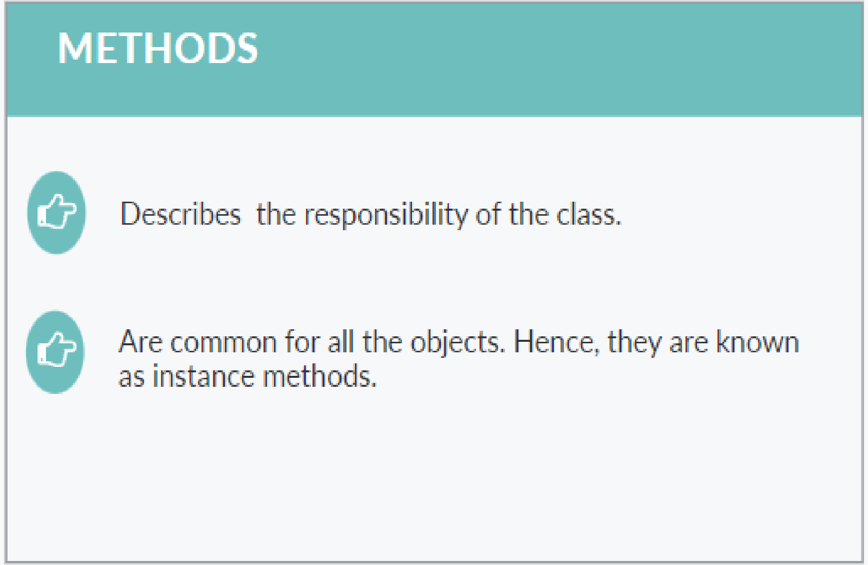
BASIC SYNTAX OF AN ATTRIBUTE

<visibility>* <modifier>* <type> <name> [= <initial_value>]

Example:

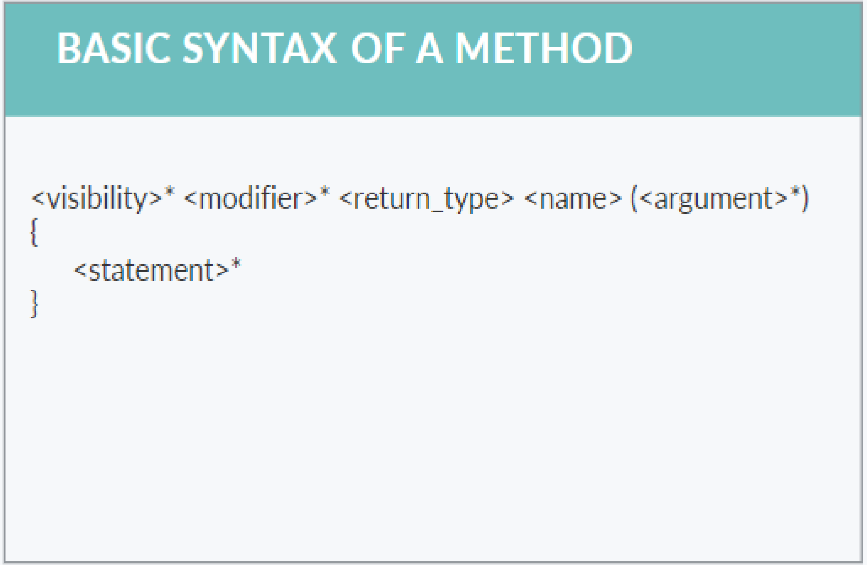
```
private double salary;
```

METHOD DECLARATION



METHODS

- Describes the responsibility of the class.
- Are common for all the objects. Hence, they are known as instance methods.

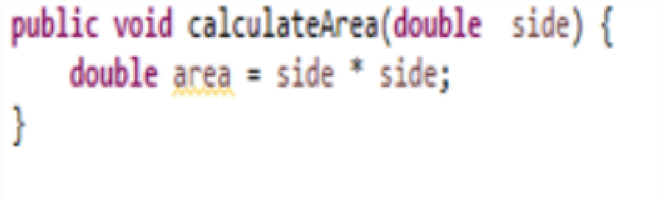


BASIC SYNTAX OF A METHOD

```
<visibility>* <modifier>* <return_type> <name> (<argument>*)
{
      <statement>*
}
```

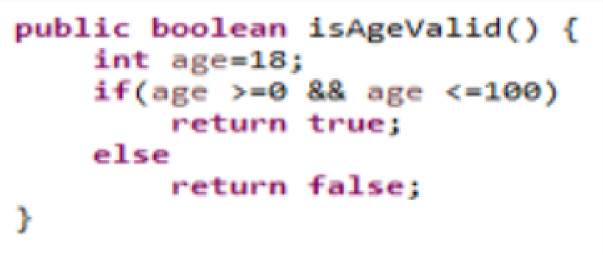
EXAMPLE

EXAMPLE 1



```
public void calculateArea(double  side) {
    double area = side * side;
}
```

EXAMPLE 2



```
public boolean isAgeValid() {
    int age=18;
    if(age >=0 && age <=100)
        return true;
    else
        return false;
}
```

OBJECT CREATION

- **Creation of an Object**

  Objects are created for a class using the keyword **new**.

  In Java, all Objects are created at run time in the **heap area**.

Syntax:

```
ClassName  objectName  = new ClassName();
```

Example:

```
Employee empOne = new Employee();
```

HANDS ON TRAINING

Doubts Time

QUIZ TIME

Quiz Time

Q1. Object can be created by ____________ keywod?

A create

B this

[✓ C] new

D default

Quiz Time

Object will be create in ________________ memory area

A stack

B method

[✓ C] heap

SUMMARY

SUMMARY

- ✓ Explain Class and Object
- ✓ Create Class and Object
- ✓ Use appropriate Access Specifier
- ✓ Write Getters, Setters Method