Programming Fundamentals

Inheritance: Introduction

Produced By:
Dr. Siobhán Drohan, Ms Mairead Meagher, Ms Siobhan Roche

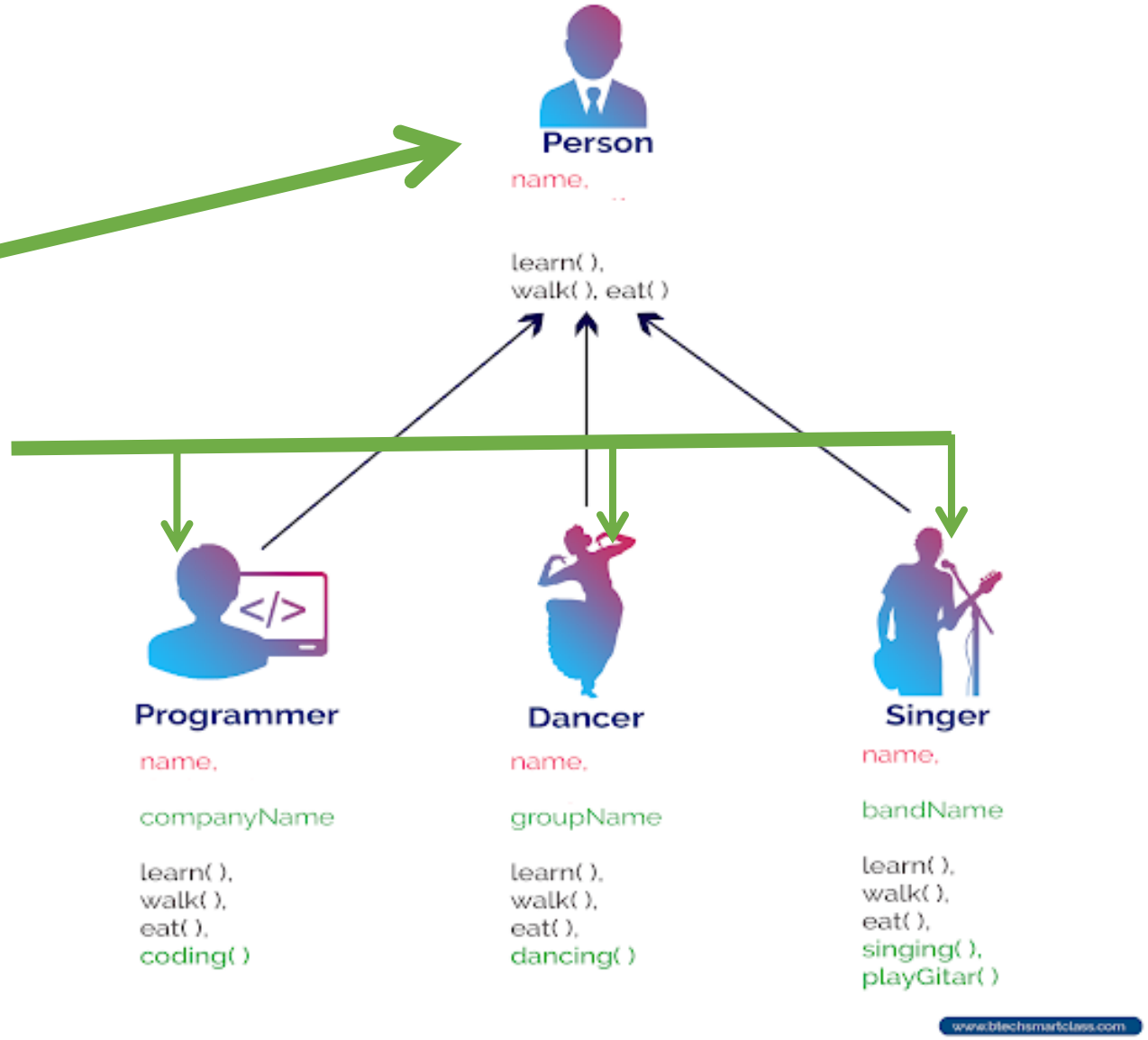
SETU
Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

Introduction to Inheritance in Java

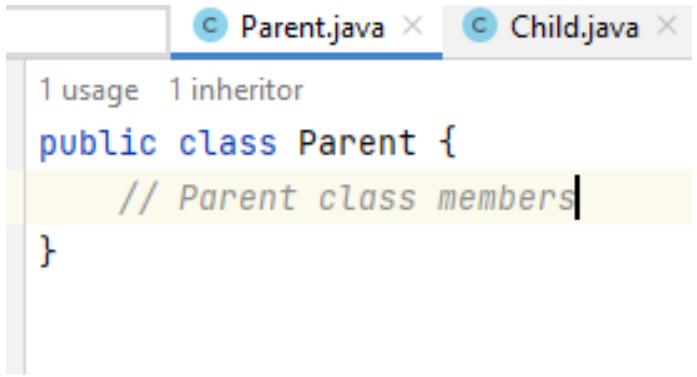
- Inheritance is a fundamental feature of object-oriented programming (OOP) that allows a class to inherit properties and behaviours from another class.
- It promotes code reusability, extensibility, and organisation by allowing new classes to be built upon existing classes

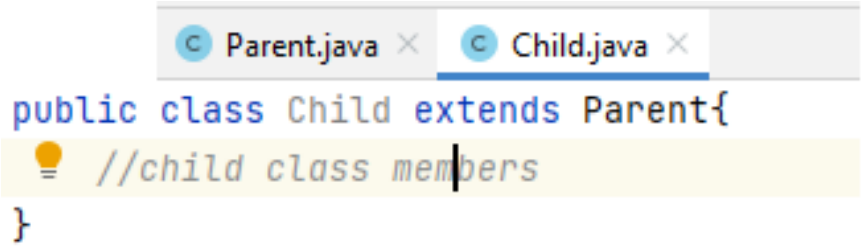
Basic Concept

- **Parent Class (Superclass/Base Class):**
  - Defines common properties and behaviours shared by multiple classes.
  - Also referred to as superclass or base class.
- **Child Class (Subclass/Derived Class):**
  - Inherits properties and behaviours from the parent class.
  - Can have its own additional properties and behaviours.
  - Also referred to as subclass or derived class.



Syntax for Inheritance in Java





extends

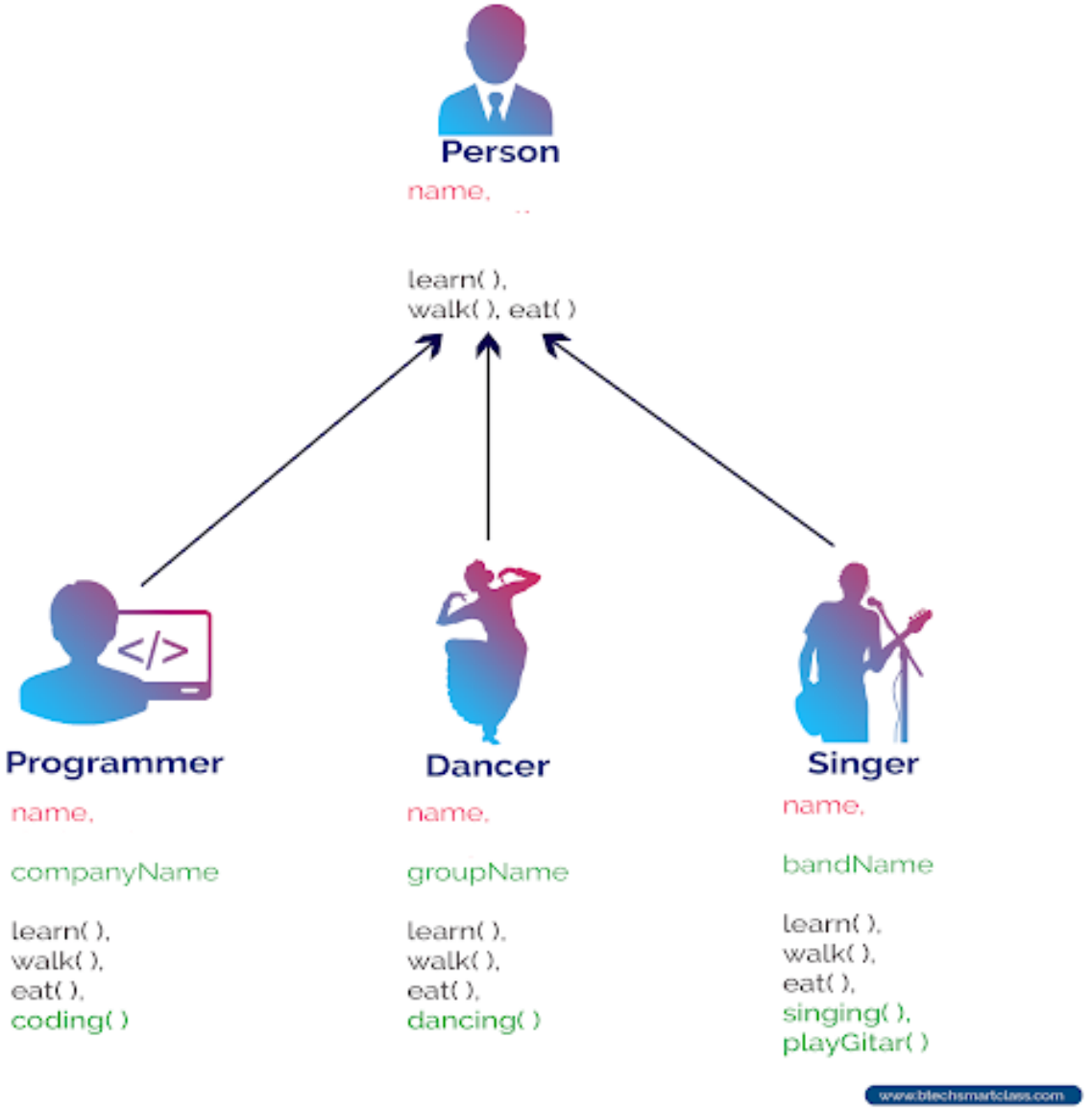
Indicates that a class is inheriting from another class

Example - Person

```java
public class Person {
    2 usages
    String name;

    public Person(String name) {
        this.name = name;
    }

   //setters and getters
    public void learn(){
        System.out.println("Learns loads");
    }
    public void eat(){
        System.out.println("Eats too much");
    }
    public void walk(){
        System.out.println("Probably doesn't walk enough");
    }
    @Override
    public String toString() {
        return "Person{" +
                "name='" + name + '\'' +
                '}';
    }
}
```

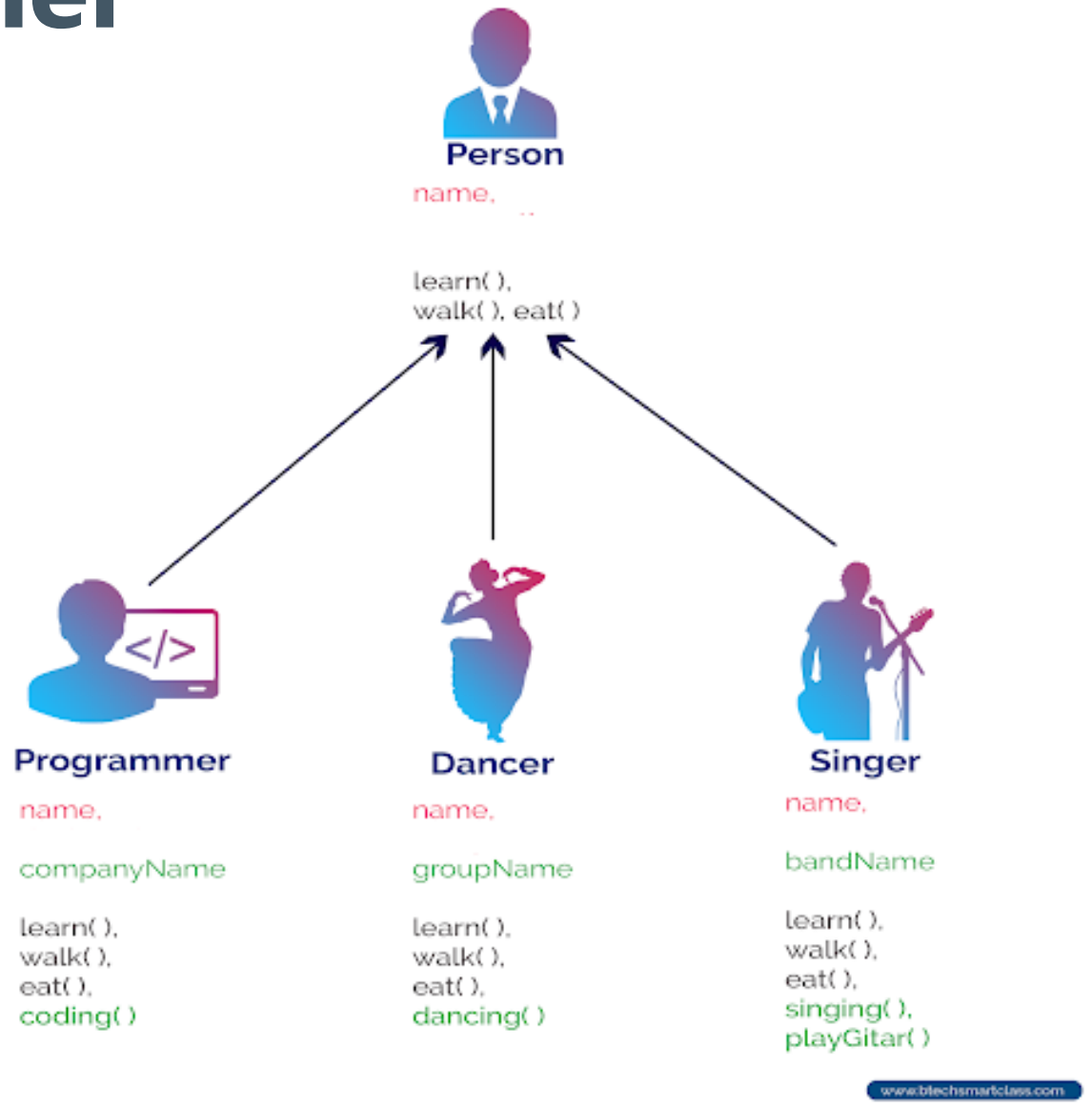


Example - Programmer

```java
public class Programmer extends Person{
    2 usages
    String companyName;

    public Programmer(String name, String companyName) {
        super(name);
        this.companyName = companyName;
    }
    //setters getters
    public void coding(){
        System.out.println("Who doesn't love to be able to code??");
    }

    @Override
    public String toString() {
        return super.toString() + "Programmer{" +
                "companyName='" + companyName + '\'' +
                '}';
    }
}
```

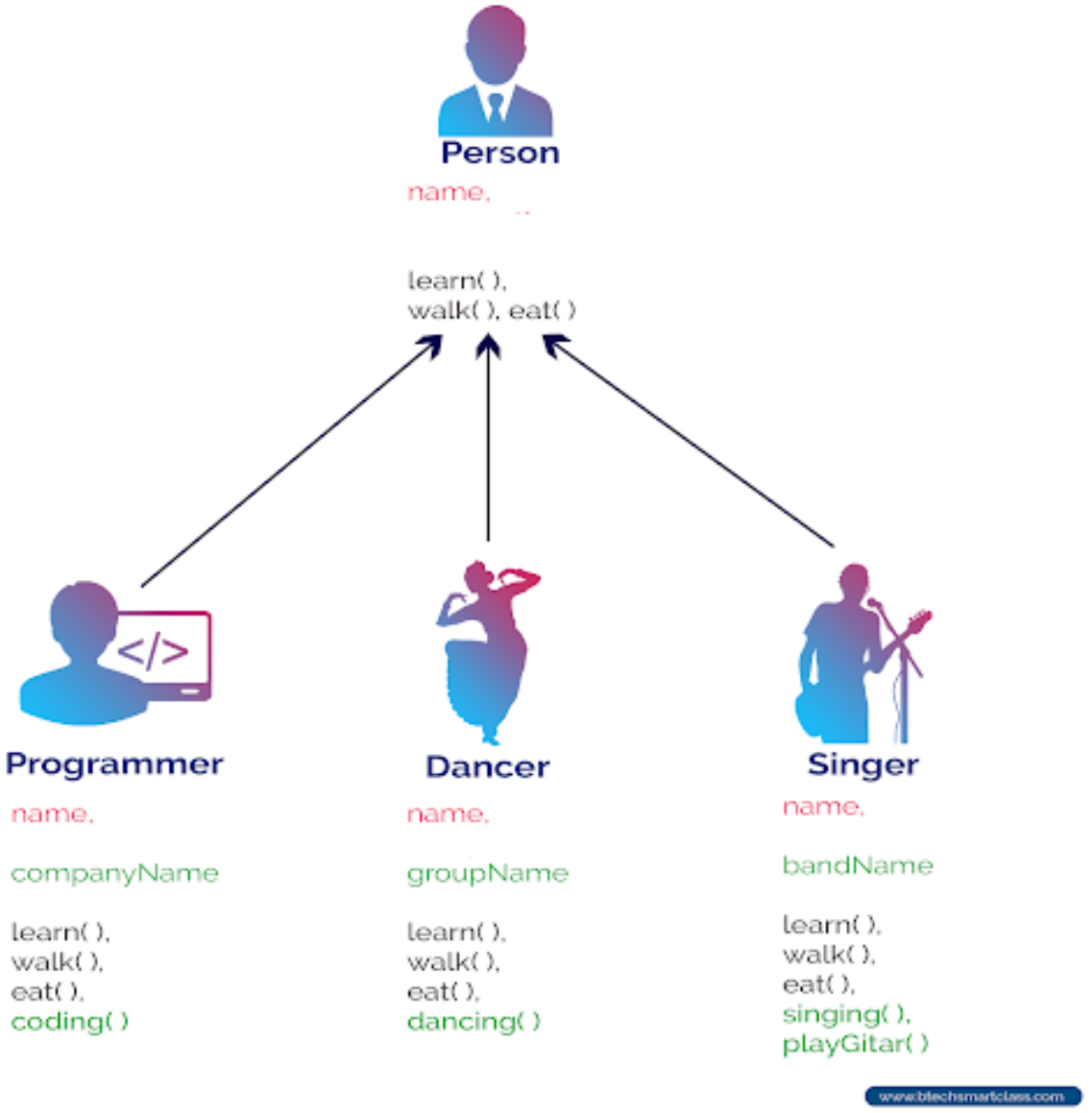


Example - Dancer

```java
public class Dancer extends Person{
    2 usages
    String groupName;

    public Dancer(String name, String groupName) {
        super(name);
        this.groupName = groupName;
    }
    //setters getters
    public void dancing(){
        System.out.println("I love to dance!!!!");
    }

    @Override
    public String toString() {
        return super.toString() + "Dancer{" +
                "groupName='" + groupName + '\'' +
                '}';
    }
}
```

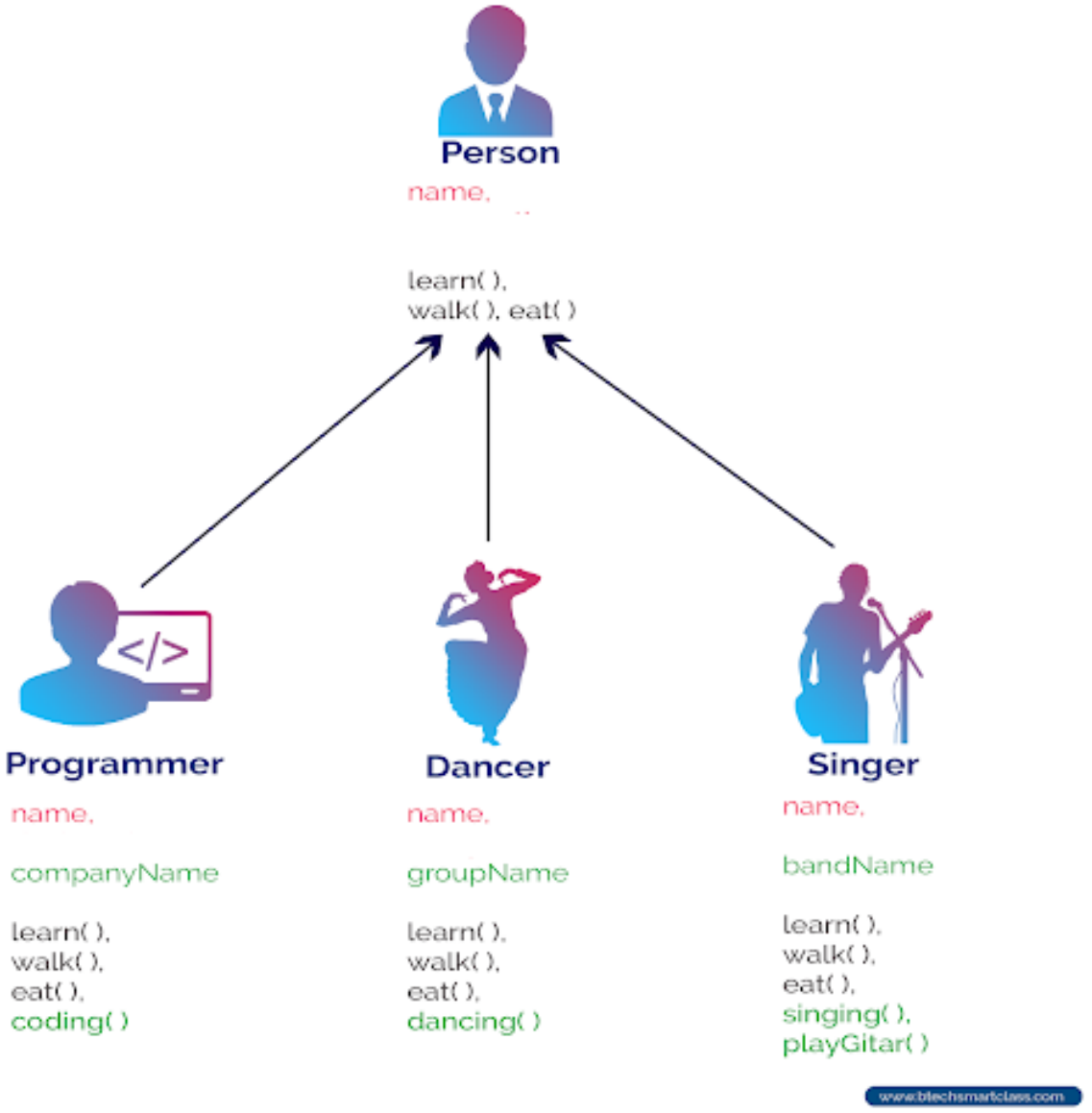


Example - Singer

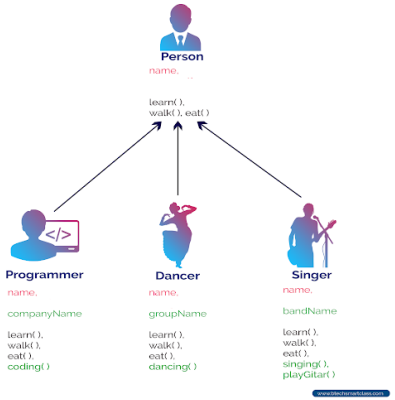
```java
public class Singer extends Person{
    2 usages
    String bandName;

    public Singer(String name, String bandName) {
        super(name);
        this.bandName = bandName;
    }
    //setters getters
    public void singing(){
        System.out.println("I love to sing!!!!");
    }
    public void playGuitar(){
        System.out.println("Only when I sing");
    }

    @Override
    public String toString() {
        return "Singer{" +
                "bandName='" + bandName + '\'' +
                ", name='" + name + '\'' +
                '}';
    }
}
```



Example - Driver




```java
public class Driver {
    public static void main(String[] args){
        Programmer progObject = new Programmer( name: "Joe Doe",  companyName: "Apple");
        Singer singerObject = new Singer( name: "Debbie Harry",  bandName: "Blondie");
        Dancer dancerObject = new Dancer( name: "Anna Pavlova",  groupName: "Imperial Russian Ballet");

        System.out.println(progObject.toString());
        System.out.println("What can the dancer do?" );
        dancerObject.eat();
        dancerObject.dancing();
        System.out.println("How about the Singer?" );
        singerObject.walk();
        singerObject.playGuitar();

    }
}
```

```
Person{name='Joe Doe'}Programmer{companyName='Apple'}
What can the dancer do?
Eats too much
I love to dance!!!!
How about the Singer?
Probably doesn't walk enough
Only when I sing
```

Any Questions?