What is association in java:

In Java, an association is a relationship between two or more classes, where objects of one class are related to objects of another class. It represents a "has-a" relationship, indicating that one class has a reference to another class. Associations are typically implemented through instance variables or method parameters.

There are three types of associations in Java:

1. **One-to-One Association:** This association represents a relationship where one object of a class is associated with exactly one object of another class. It is implemented using a reference variable.

Example:

```
class Person {
   private Address address;
   // ...
}

class Address {
   // ...
}
```

2. **One-to-Many Association:** This association represents a relationship where one object of a class is associated with multiple objects of another class. It is typically implemented using a collection (e.g., ArrayList) or an array.

Example:

```
class Department {
   private List<Employee> employees;
   // ...
}

class Employee {
   // ...
}
```

3. **Many-to-Many Association:** This association represents a relationship where multiple objects of one class are associated with multiple objects of another class. It is implemented using collections or arrays in both classes.

Example:

```
class Student {
    private List<Course> courses;
    // ...
}

class Course {
    private List<Student> students;
    // ...
}
```

Associations can be unidirectional or bidirectional. In a unidirectional association, only one class has a reference to another class, while in a bidirectional association, both classes have references to each other.

To establish an association between classes, you can either set the reference variables directly or use methods to add, remove, or retrieve associated objects.

Associations allow classes to collaborate and exchange information, facilitating complex relationships and enhancing the flexibility and reusability of the code. They are an essential part of object-oriented programming and enable the modeling of real-world scenarios.

In Java, there are there are two main types of associations that can exist between classes:

1. **Aggregation:** Aggregation is a type of association where one class has a "has-a" relationship with another class, but the associated class can exist independently. It represents a whole-part relationship. The associated class can be shared by multiple instances of the main class. Aggregation is represented by a diamond-shaped arrow pointing from the main class to the associated class.

Example:

```
class Department {
    private List<Employee> employees;
    // ...
}
```

2. **Composition:** Composition is a stronger form of aggregation where the associated class is part of the main class and cannot exist independently. If the main class is destroyed, the associated class is also destroyed. Composition is represented by a filled diamond-shaped arrow pointing from the main class to the associated class.

Example:

```
class Car {
    private Engine engine;
    // ...
}
```

Examples in class

Aggregation demo:

```
package util.aggregation;

public class MobilePhone {
    private int number;

    public MobilePhone(int number) {
        this.number = number;
    }

    public int getNumber() {
        return number;
    }

    public void setNumber(int number) {
        this.number = number;
    }

    @Override
    public String toString() {
        return "MobilePhone{" +
                "number=" + number +
                '}';
    }
}
```

```
package util.aggregation;

public class Person {
    private MobilePhone mobilePhone;

    public Person(){

    }

    public MobilePhone getMobilePhone() {
        return mobilePhone;
    }
```

```java
    public void setMobilePhone(MobilePhone mobilePhone) {
        this.mobilePhone = mobilePhone;
    }

    @Override
    public String toString() {
        return "Person{" +
                "mobilePhone=" + mobilePhone +
                '}';
    }
}

package util.aggregation;

public class AggregationDemo {


    public static void main(String[] args) {
        Person person = new Person();
        MobilePhone mobilePhone = new MobilePhone(12345);

        System.out.println(person);

        person.setMobilePhone(mobilePhone);

        System.out.println(person);
    }

}
```

Association demo:

```java
package util.association;

public class Heart {
    private int beatsPerMinutes;
    private int bloodFlow;
    private Human human;

    public Heart(int beatsPerMinutes, int bloodFlow) {
        this.beatsPerMinutes = beatsPerMinutes;
        this.bloodFlow = bloodFlow;
    }

    public int getBeatsPerMinutes() {
        return beatsPerMinutes;
    }

    public void setBeatsPerMinutes(int beatsPerMinutes) {
        this.beatsPerMinutes = beatsPerMinutes;
    }

    public int getBloodFlow() {
        return bloodFlow;
    }

    public void setBloodFlow(int bloodFlow) {
        this.bloodFlow = bloodFlow;
    }

    public Human getHuman() {
        return human;
    }

    public void setHuman(Human human) {
        this.human = human;
    }

    @Override
    public String toString() {
        return "Heart{" +
                "beatsPerMinutes=" + beatsPerMinutes +
                ", bloodFlow=" + bloodFlow +
                '}';
    }
}
```

```
package util.association;

public class Human {
    public Heart heart;

    public Human() {
        this.heart = new Heart(72, 100);
        this.heart.setHuman(this);
    }

    public Heart getHeart() {
        return heart;
    }

    @Override
    public String toString() {
        return "Human{" +
                "heart=" + heart +
                '}';
    }
}
```

```
package util.association;

public class AssociationDemo {
    public static void main(String[] args) {
        Human human = new Human();
        System.out.println(human);
    }
}
```