16-'this' keyword

The 'this' Keyword in Java

In Java, the '**this**' keyword is a reference variable that refers to the current object, meaning it points to the object instance on which the method or constructor is being invoked. It can be used in various ways to improve code readability and avoid naming conflicts between local variables and instance variables.

Key Uses of this Keyword:

1. **Refer to Current Class Instance Variables**:
    - When a local variable and an instance variable have the same name, this helps to differentiate between them.
2. **Invoke Current Class Methods**:
    - You can use this to call another method of the current class.
3. **Return the Current Class Instance**:
    - Useful in method chaining or builder pattern.
4. **Pass the Current Class Instance as a Parameter**:
    - You can pass the current object as an argument in a method or constructor.

Example Code

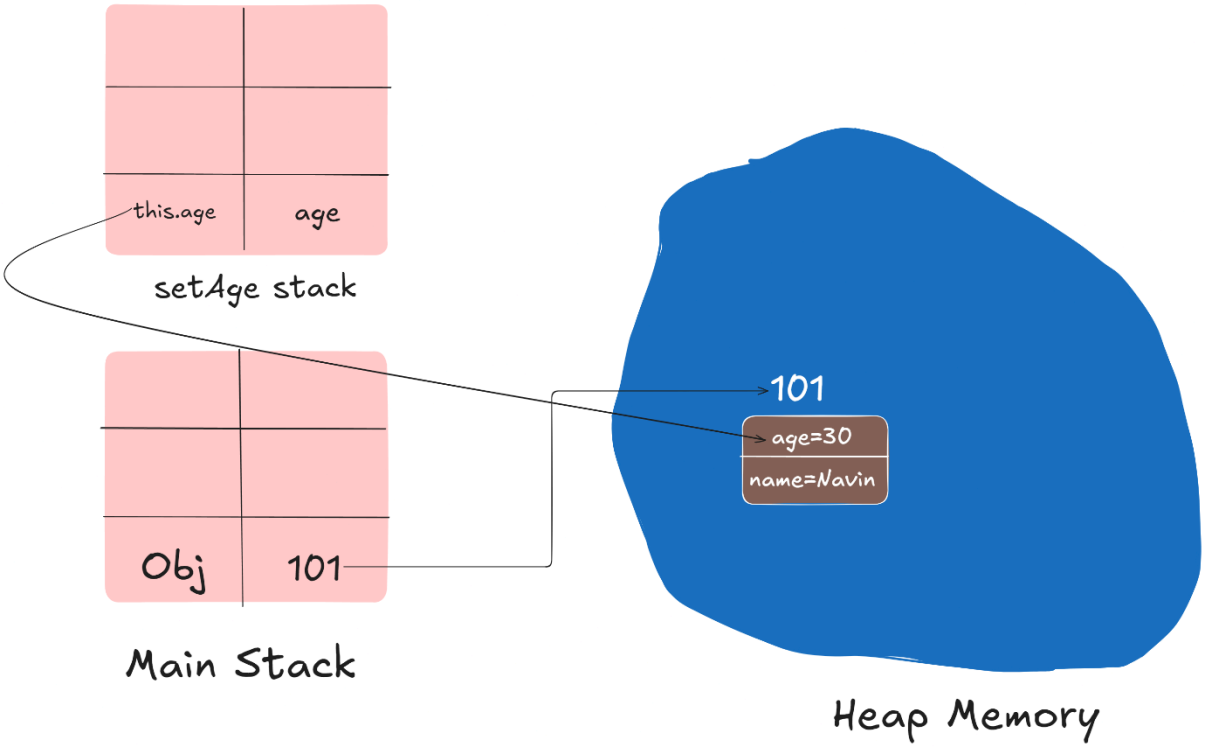
```java
class Human {
    private int age;
    private String name;

    public int getAge() {
        return age;
    }

    public String getName() {
        return name;
    }

    public void setAge(int age) {
        this.age = age; // 'this' refers to the current object
    }

    public void setName(String name) {
        this.name = name; // 'this' refers to the current object
    }
}

public class Demo {
    public static void main(String[] args) {
        Human obj = new Human();
        obj.setAge(30);
        obj.setName("Navin");
        System.out.println("Name: " + obj.getName() + "\nAge: " + obj.getAge());
    }
}

```

Explanation of the Code:

- **Class Human**: This class has two private instance variables, age and name. It also has getter and setter methods.
- **setAge(int age) and setName(String name) Methods**: The this keyword is used to assign the method's local variables (age and name) to the instance variables of the same name.
- **Class Demo**: This class creates an instance of the Human class, sets values using setters, and retrieves them using getters.

Output:

If we did not use this in the setter methods, the instance variables would not be correctly assigned, leading to potential bugs or unexpected results.



Methods to Use this in Java:

1. **Using this to Refer to Current Class Instance Variables**:
    - Differentiates between local and instance variables when they share the same name.
2. **Using this to Invoke Current Class Methods**:
    - Calls another method in the same class.
3. **Using this to Return the Current Class Instance**:
    - Facilitates method chaining.
4. **Using this as a Method Parameter**:
    - Passes the current object as an argument to another method or constructor.

Disadvantages of Using this:

1. **Overuse Can Reduce Code Readability**:
    - Excessive use may clutter the code and make it harder to follow.
2. **Unnecessary Overhead**:
    - Using this when not needed can add unnecessary complexity.
3. **Cannot Use this in Static Context**:
    - The '**this**' keyword cannot be used in a static method because static methods belong to the class, not any specific object.