Here are detailed notes based on the document about **Super and This Keywords in Java**:

Super Keyword in Java

1. **Definition**:
   - `super` is a reference variable that refers to the immediate superclass object.
   - Used in the context of inheritance to access members of the superclass from a subclass.
2. **Key Features**:
   - Represents a superclass object.
   - When an object of a subclass is created, an implicit object of the superclass is also created, referred to by `super`.
   - Enables access to superclass variables, methods, and constructors.
3. **Usage**:
   - To explicitly call superclass members (variables, methods).
   - Implicitly used by the JVM to call the default constructor of the superclass from a subclass constructor.
4. **Advantages**:
   - Resolves ambiguity when both superclass and subclass have members with the same name.
   - Facilitates **constructor chaining**:
     - If no constructor call is explicitly made, `super()` is automatically added by the Java compiler.
   - Useful in overriding:
     - The `super` keyword can call the overridden method in the superclass.
   - Enables calling specific constructors of the superclass with `super(arguments)`.
   - Maintains inheritance hierarchy and clarifies access to superclass members.

This Keyword in Java

1. **Definition**:
   - `this` is a reference variable referring to the current class object.
   - Used to distinguish between the instance variables and parameters or local variables.
2. **Key Features**:
   - Refers to the current class object in instance methods, constructors, and instance blocks.
   - Applicable only within non-static contexts (methods or blocks).
3. **Syntax Examples**:
   - **Accessing instance variables**:

```
this.a;
```

- **Calling current class methods**:

```
this.msg();
```

- **Invoking current class constructors**:

```
this(arguments);
```

4. **Usage**:
   - **Access Instance Variables**: Resolves naming conflicts between instance variables and parameters.
   - **Call Methods**: Invokes current class non-static methods.
   - **Invoke Constructors**: Calls another constructor of the same class using `this()`.
   - **Parameter Usage**:
     - Can be passed as a parameter in method or constructor calls.
   - **Return Current Object**: Returns the current class object from a method.

Differences Between `Super` and `This` Keywords

Aspect	`this` Keyword	`super` Keyword
Reference	Refers to the current class object.	Refers to the immediate superclass object.
Scope	Accesses current class members.	Accesses superclass members, even if overridden.
Constructor Usage	Calls another constructor in the same class.	Calls the superclass constructor from the subclass.
Default Behavior	Java compiler does not insert `this()` automatically.	Compiler automatically adds `super()` if no call is made.

General Notes

1. Neither `this` nor `super` can be used inside static methods or static initialization blocks.
2. Both are essential for managing inheritance and object references efficiently.

These notes provide a comprehensive understanding of the **Super** and **This** keywords in Java. Let me know if you'd like further elaboration or examples!