10.3-Anonymous Inner class

Definition:
In Java, an anonymous inner class is a type of inner class without a name. It is used to create an object with certain modifications or behaviours without explicitly declaring a subclass. This feature is commonly used when you need a one-time use implementation for a class or an interface.

Key Points:

- **Anonymous** means without a name.
- **Anonymous inner classes** are created without naming the class.
- They are typically used to extend a class or implement an interface.

Syntax

The syntax for creating an anonymous inner class involves the following:

```
new ClassName() {
    // method overriding or additional methods
};
```

Example 1: Without Anonymous Inner Class

```
class A {
    public void show() {
        System.out.println("In A's show method");
    }
}

class B extends A {
    @Override
    public void show() {
        System.out.println("In B's show method");
    }
}

public class Demo {
    public static void main(String[] args) {
        A obj = new B();  // B class object
        obj.show();
    }
}
```

Output:

```
In B's show method
```

Explanation:

- In this example, we have a class A with a method show().
- Class B extends A and overrides the show() method.
- The object obj is created for class B, which results in the overridden method show() being called.
- This works but requires creating a separate class (B) even if it's only used once.

Example 2: Using Anonymous Inner Class

```
class A {
    public void show() {
        System.out.println("In A's show method");
    }
}

public class Demo {
    public static void main(String[] args) {
        A obj = new A() {
            @Override
            public void show() {
                System.out.println("In new anonymous show method");
            }
        };
        obj.show();
    }
}
```

Output:

```
In new anonymous show method
```

Explanation:

- Instead of creating a separate subclass like B, we use an anonymous inner class.

- The curly braces {} after new A() define the anonymous inner class, where we override the show() method.
- This creates a class without a name, allowing a one-time customization of A.

Advantages of Anonymous Inner Classes:

1. **Encapsulation**: Keeps implementation details close to where they are used, improving code organization.
2. **Access Control**: Can access private members of the outer class, even if they are private.
3. **Code Optimization**: Avoids the need to create a separate class, reducing boilerplate code.
4. **Polymorphism**: Provides a dynamic way to override methods at runtime.
5. **Reduced Code Complexity**: Simplifies the code when you need minor changes to a class or interface just once.

When to Use:

- When you need to override methods of a class or interface just once.
- When a specific implementation is required without creating a new, named class.

Key Features of Anonymous Inner Class:

- **No name**: It is defined without any class name.
- **Single Use**: It's typically used for one-time code modifications.
- **Extends or Implements**: You can extend a class or implement an interface on the fly.