Problem 1: Basic Inner Class Access

Problem: Create a class `Outer` that has an inner class `Inner`. The `Inner` class should have a method `showMessage()` that prints "Hello from Inner class!". Create an object of `Outer` and use it to instantiate `Inner` and call the `showMessage()` method.

Solution:

```
java
Copy code
class Outer {
    class Inner {
        public void showMessage() {
            System.out.println("Hello from Inner class!");
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Outer outer = new Outer();  // Create outer class object
        Outer.Inner inner = outer.new Inner();  // Create inner class object
        inner.showMessage();  // Call inner class method
    }
}
```

Output:

```
csharp
Copy code
Hello from Inner class!
```

Problem 2: Static Inner Class Example

Problem: Create a class `Outer` with a static inner class `StaticInner` that has a method `display()`, which prints "This is a static inner class". Call this method from the `main()` method without creating an instance of `Outer`.

Solution:

```
java
Copy code
class Outer {
    static class StaticInner {
        public void display() {
            System.out.println("This is a static inner class");
        }
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Outer.StaticInner inner = new Outer.StaticInner();  // No need for
Outer object
        inner.display();  // Call static inner class method
    }
}
```

Output:

```
csharp
Copy code
This is a static inner class
```

Problem 3: Accessing Outer Class Members from Inner Class

Problem: Create a class `Outer` that has an integer field `x = 10`. The inner class `Inner` has a method `printX()` that accesses and prints the value of `x`. Demonstrate how the inner class can access the outer class's members.

Solution:

```
java
Copy code
class Outer {
    private int x = 10;  // Outer class field

    class Inner {
        public void printX() {
            System.out.println("Outer x = " + x);  // Access outer class
field
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Outer outer = new Outer();
        Outer.Inner inner = outer.new Inner();
        inner.printX();  // Output the value of x
    }
}
```

Output:

```
java
Copy code
Outer x = 10
```

Problem 4: Local Inner Class Example

Problem: Create a method in class `Outer` called `useLocalClass()` which defines a local inner class `LocalInner` inside it. The `LocalInner` class should have a method `printMessage()` that prints "Hello from Local Inner Class!". Call this method inside `useLocalClass()`.

Solution:

```
java
Copy code
class Outer {
    public void useLocalClass() {
        class LocalInner {
            public void printMessage() {
                System.out.println("Hello from Local Inner Class!");
            }
        }

        LocalInner local = new LocalInner();  // Create instance of local
inner class
        local.printMessage();  // Call method of local inner class
    }
}

public class Main {
    public static void main(String[] args) {
        Outer outer = new Outer();
        outer.useLocalClass();  // Use local inner class
    }
}
```

Output:

```
sql
Copy code
Hello from Local Inner Class!
```

Problem 5: Accessing Final Variables in Local Inner Class

Problem: In class `Outer`, create a method `printValue()` that takes an integer `value` as an argument. Inside this method, define a local inner class that has a method `print()` which prints the value of `value`. The value of `value` should be accessible in the local inner class.

Solution:

```
java
Copy code
class Outer {
    public void printValue(final int value) {  // Value must be final or
effectively final
        class LocalInner {
            public void print() {
```

```
                System.out.println("Value = " + value);  // Access final
variable
            }
        }

        LocalInner local = new LocalInner();
        local.print();
    }
}

public class Main {
    public static void main(String[] args) {
        Outer outer = new Outer();
        outer.printValue(42);  // Pass a value to the method
    }
}
```

Output:

```
makefile
Copy code
Value = 42
```