Scopes

Java™

What are Scopes

- Scope refers to the visibility of a class, field, variable or method.
- A scope is any enclosed code inside curly bracers {}

Local Variables and Objects

```
public static void main(String[] args) {
    int result = 0;

    for(int i=0; i<5; i++) {
        result = result + i;
    }

    System.out.println(result);
    System.out.println(i);
}
```

Result is visible in main and all subscopes

Int i is visible in the for loop and all subscopes

Will print 10 since result is visible in main

Will cause a compile error

Local Variables and Objects

```
public static void main(String[] args) {
    int result = 0;

    for(int i=0; i<5; i++) {
        result = result + i;
    }

    System.out.println(result);
    print();
}

public static void print() {
    System.out.println(result);
}
```

Compile Error

Local Variables are only visible to the method or scope it was created and subscopes of that scope.

Local Variables and Objects

```
public static void main(String[] args) {
    int result = 0; //OK

    {
        int result = 1; //Compile Error
    }
}

public static void print() {
    int result = 2; //OK
}
```

Local variables needs to have unique names only for its current scope and subscope

Class Fields and Internal Scopes

```
public class ScopeCheck {

    public int number = 0;
    private static int number2 = 0;

    public void print() {
        int number = 2;
        System.out.println(number);
    }

    public static void staticPrint() {
        int number2 = 2;
        System.out.println(number2);
    }

}
```

When using same name as instance field `number` the local declared variable gets used

Same with the static field, the local variable gets used.

In a class scope Java will always use most local variable. This can be avoided by adding `this.number` for instace field or `ScopeCheck.number2` for static fields

```java
public class ScopeCheck {

    public int number = 0;
    private static int number2 = 0;

    public void print() {
        int number = 2;
        System.out.println("Field: " + this.number + " LocalVariable: " + number);
    }

    public static void staticPrint() {
        int number2 = 2;
        System.out.println("Field: " + ScopeCheck.number2 + " LocalVariable: " + number2);
    }

}
```