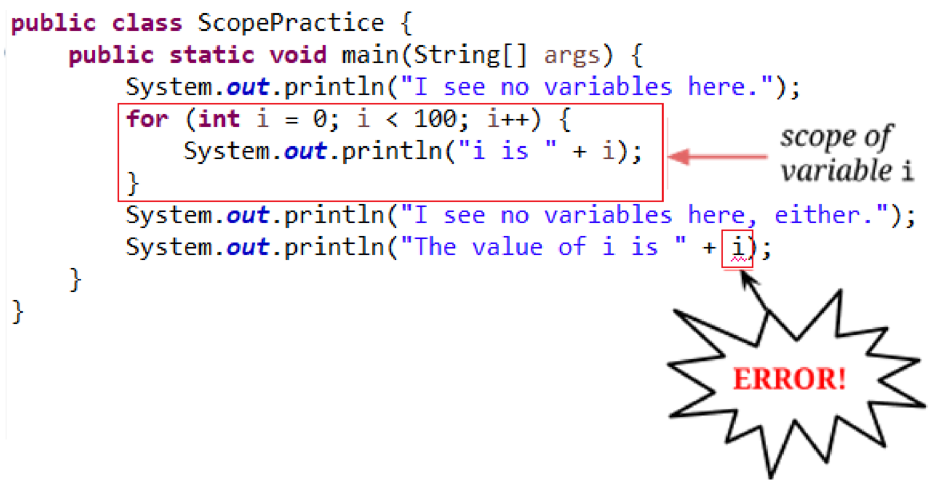
Managing Complexity

Scope

The scope of a variable is defined as the part of the program in which the variable is valid (where you are allowed to use it). Java has rules to keep scope as small as possible, because this usually makes programs less complex.

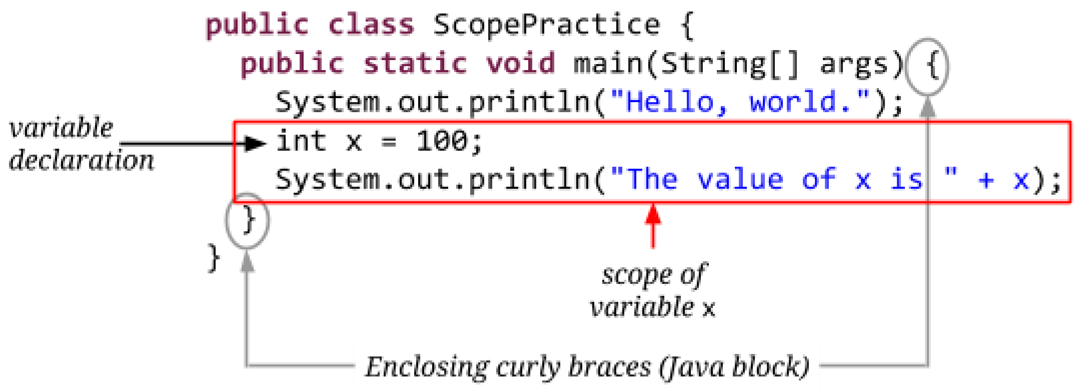


```
public class ScopePractice {
    public static void main(String[] args) {
        System.out.println("I see no variables here.");
        for (int i = 0; i < 100; i++) {
            System.out.println("i is " + i);
        }
        System.out.println("I see no variables here, either.");
        System.out.println("The value of i is " + i);
    }
}
```

For `for` loops, the scope is relatively straightforward: any variable declared in the initialize part of the for loop is valid only inside the for loop itself.

Determining the scope of other variable declarations requires more processing:

- Work backward from the variable declaration to find the first left curly brace symbol “`{`”, then find the matching right curly brace “`}`”.
- The variable’s scope is the section of the Java program starting at the variable declaration and ending at the right curly brace you just found.



```java
public class ScopePractice {
  public static void main(String[] args) {
    System.out.println("Hello, world.");
    int x = 100;
    System.out.println("The value of x is " + x);
  }
}
```

There are some important rules about scope:

- Attempting to use the variable outside of the variable’s scope is in an error.
- It is also an error to use a variable from one method in a different method, even if the second method was called from inside the variable’s scope.
- You can’t have two variables with the same name with overlapping scope.
- You can have two variables with the same name as long as their scopes don’t overlap.

Exercise 1. Identifying Scope

For the following exercises, we'll be working with this program:

```
public class ScopePractice {
     public static void fancyMethod() {
          int d = 7;
          System.out.println("d is " + d);
     }

     public static void main(String[] args) {
          int c = 10;
          fancyMethod();
          int b = 100;
          System.out.println("c is " + c);
          anotherFancyMethod();
          System.out.println("b is " + b);
     }

     public static void anotherFancyMethod() {
          int a = 1;
          for (int i = 0; i < 3; i++) {
               int e = 42;
               System.out.println("e is " + e);
          }
          System.out.println("a is " + a);
     }
}
```

Problem 1a.

Draw a box around the scope of the variable a and label the box.

Problem 1b.

Draw a box around the scope of the variable b and label the box.

Problem 1c.
Draw a box around the scope of the variable c and label the box.

Problem 1d.
Draw a box around the scope of the variable d and label the box.

Problem 1e.
Draw a box around the scope of the variable e and label the box.

Problem 1f.
What is the output of this program? Write it below.

Exercise 2. Identifying Scope Errors

Each of the following code samples has exactly one scope error. Find the error and write down why it is an error (what rule of scope it breaks.)

Problem 2a.

```
     public static void main(String[] args) {
         for (int i = 0; i < 10; i++) {
           int squared = i * i;
           System.out.println(i + " squared is " +
squared);
         }
         System.out.println("The last one was " + squared);
     }
```

Problem 2b.

```
public static void interestingMethod() {
     int x = 1;
     int y = 2;
     multiplyPlease();
}

public static void multiplyPlease() {
     System.out.println("x times y is " + (x * y));
}
```

Problem 2c.

```
     public static void main(String[] args) {
          System.out.println("About to do the 
computation!");
          doComputation();
          System.out.println(
               "The result of the computation is " + x);
     }

     public static void doComputation() {
          int x = 1;
          for (int i = 0; i < 100; i++) {
               x += 3;
               x *= 2;
          }
     }
```

Problem 2d.

```
     public static void scopeIsFun() {
         int i = 0;
         System.out.println("Let's get started.");
         for (int i = 0; i < 10; i++) {
            System.out.println("My favorite number is " +
i);
         }
         System.out.println(
             "I guess I have a lot of favorite numbers!");
     }
```

Problem 2e.

```
     public static void almostDone() {
         int i = 0;
         for (j = 0; j < 10; j++) {
             System.out.println(
                 "My favorite number is still " + j);
         }
         System.out.println(
            "I guess I still have a lot of favorite
numbers!");
     }
```

Problem 2f.

```
     public static void finallyTheLastProblemHooray() {
          System.out.println("I guess the answer is " + x);
          int x = 42;
     }
```