

Scope of Variables

Hui Chen

Department of Computer & Information Science

Brooklyn College

Objectives

- To determine the scope of variables (§6.9).

Outline

- Discussed
 - Defining and invoking value-returning methods
 - Defining and invoking void methods
 - Parameter passing and passing by value
 - Using method to modularize several example problems (including converting hexadecimal to decimal)
 - Method overload,
 - Pitfalls and errors (including ambiguous method invocation)
- To discuss
 - Scope of variables
 - Local variables and scope of local variables

Scope of Variables

- The part of the program where the variable can be referenced

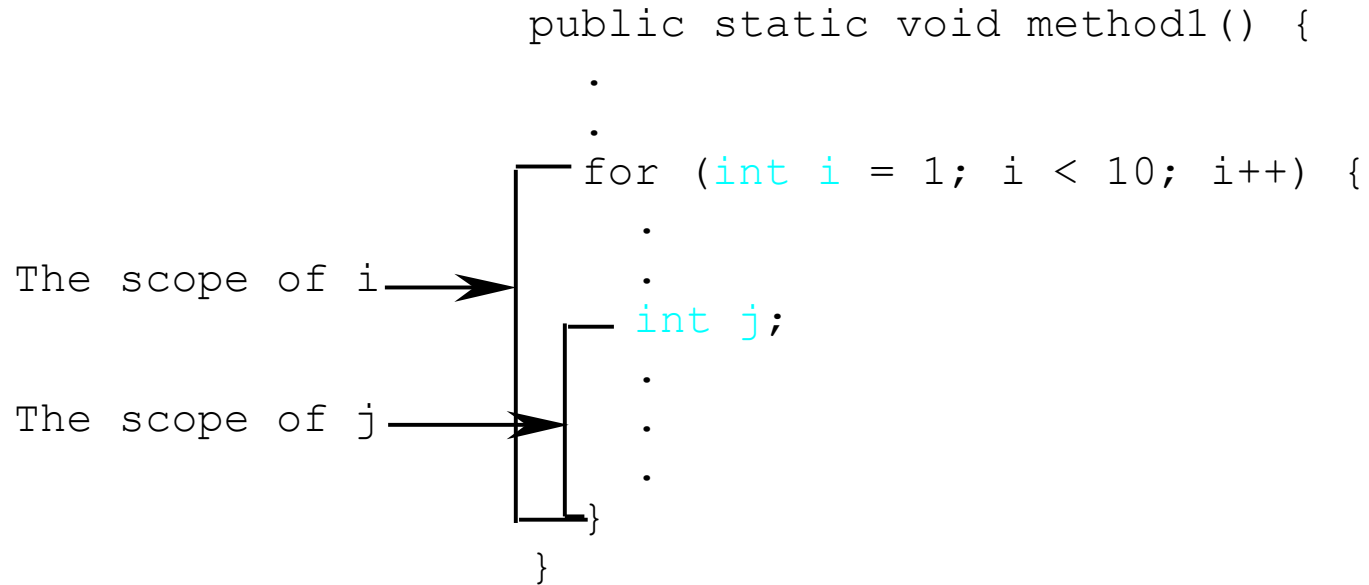
Scope of Local Variables

- A local variable: a variable defined inside a method
- The scope of a local variable starts from its declaration and continues to the end of the block that contains the variable.
- A local variable must be declared before it can be used.
- You can declare a local variable with the same name multiple times in different non-nesting blocks in a method, but you cannot declare a local variable twice in nested blocks.

Scope of Local Variables and For Loops

- A variable declared in the *initial action* part of a for loop header has its scope in the entire loop.
- But a variable declared inside a for loop body has its scope limited in the loop body from its declaration and to the end of the block that contains the variable

For Loop Example



Multiple Variables, Same Name

It is fine to declare `i` in two non-nesting blocks

```
public static void method1() {  
    int x = 1;  
    int y = 1;  
  
    for (int i = 1; i < 10; i++) {  
        x += i;  
    }  
  
    for (int i = 1; i < 10; i++) {  
        y += i;  
    }  
}
```

It is wrong to declare `i` in two nesting blocks

```
public static void method2() {  
    int i = 1;  
    int sum = 0;  
  
    for (int i = 1; i < 10; i++)  
        sum += i;  
}
```


How about this example?

Compilation error or not?

```
public static void correctMethod() {
    int x = 1;
    int y = 1;
    // i is declared
    for (int i = 1; i < 10; i++) {
        x += i;
    }
    // i is declared again
    for (int i = 1; i < 10; i++) {
        y += i;
    }
}
```

How about this example?

Compilation error or not?

```
public static void incorrectMethod() {  
    int x = 1;  
    int y = 1;  
    for (int i = 1; i < 10; i++) {  
        int x = 0;  
        x += i;  
    }  
}
```

Questions