October 2nd, 2013: Announcements

Assignment 3 is due Today at 9:00 pm. Please don’t wait until the last few hours to submit and test your code. WebGS cannot handle 400 students submitting at one time.

Assignment 4 is due next Wednesday (Oct 9nd) at 9:00 pm.

Office Hours: 1-3pm on Monday and Wednesday (4 total hours of office hours)

Sample Exam and Study Tips have been posted to D2L

Remember that you are not allowed to look at other students’ code. If you send someone your code, or receive code from another student, that is a direct violation of Academic Integrity.

Remember to complete your OELs. Log on to the site to check if you have any. Rick Snodgrass posted a note earlier addrsesing some of the problems you might be having, so check there first. If you are still having trouble post a note on Piazza.

Block Scope:

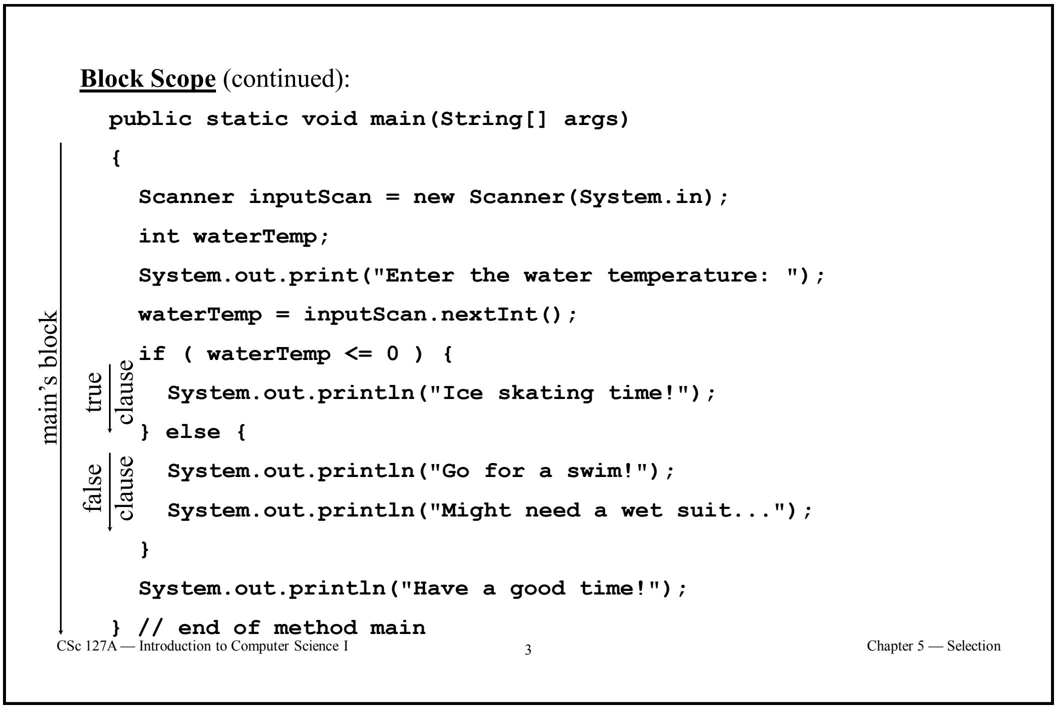
Reading: page 225.

The *scope* of a variable is the region of code within a program where the variable can be referenced (or used).

Scope is determined by the *block* of code containing the variable declaration.

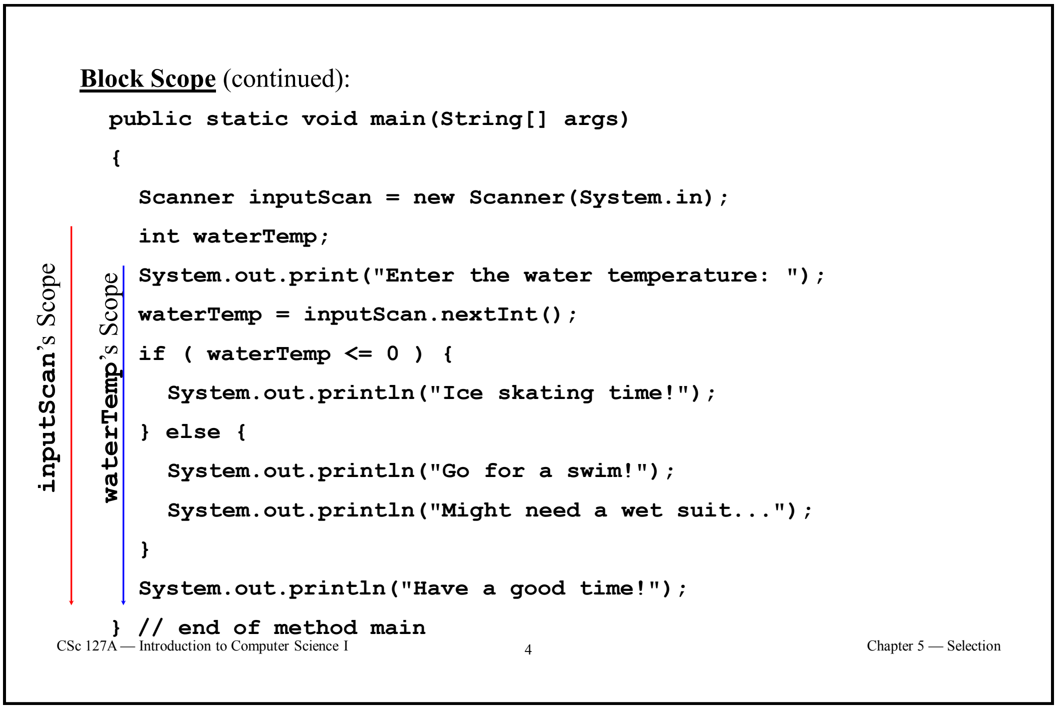
Code blocks:

- The **`main`** method is a code block.
- Code in the *true* clause of an **`if`** statement is a block.
- Code in the *false* clause of an **`if`** statement is a block.
- Code inside **`{ }`**’s is a block.



Block Scope (continued):

```
public static void main(String[] args)
{
  Scanner inputScan = new Scanner(System.in);
  int waterTemp;
  System.out.print("Enter the water temperature: ");
  waterTemp = inputScan.nextInt();
  if ( waterTemp <= 0 ) {
    System.out.println("Ice skating time!");
  } else {
    System.out.println("Go for a swim!");
    System.out.println("Might need a wet suit...");
  }
  System.out.println("Have a good time!");
} // end of method main
```



Block Scope (continued):

```
public static void main(String[] args)
{
  Scanner inputScan = new Scanner(System.in);
  int waterTemp;
  System.out.print("Enter the water temperature: ");
  waterTemp = inputScan.nextInt();
  if ( waterTemp <= 0 ) {
    System.out.println("Ice skating time!");
  } else {
    System.out.println("Go for a swim!");
    System.out.println("Might need a wet suit...");
  }
  System.out.println("Have a good time!");
} // end of method main
```

<u>Block Scope</u> (continued):

This code works. There are two distinct variables, both named **`area`**. Each exists in a different scope.

```
 public static void main(String[] args)
 {
   Scanner inputScan = new Scanner( System.in );
   System.out.print("Enter the width and height: ");
   int width = inputScan.nextInt();
   int height = inputScan.nextInt();
   if ( width == height ) {
       int area = width * width;
area   System.out.println("Area of square = " + area);
   } else {
       int area = width * height;
area   System.out.println("Area of rectangle = " + area);
   }
 } // end of method main
```

<u>Block Scope</u> (continued):

This code does not work. There are (still) two distinct variables, both named **`area`**. Each exists in a different scope. The reference to **`area`** after the **`if`** is invalid, since it lies outside the scope of both **`area`** variables.

```
//to save room I removed the lines of code that get
//width and height as use input
if ( width == height ) {
   int area = width * width;
area System.out.println("Area of square = " + area);
} else {
   int area = width * height;
area System.out.println("Area of rectangle = " + area);
}
int inchesArea;
inchesArea = area * 144;  // 144 sq inches in 1 sq foot
System.out.println("Area in square inches = " +
                    inchesArea);
} // end of method main
```

Compiler Error: cannot find symbol. How can this be fixed?

Testing Techniques:

Reading: Section 5.6

Execution Path Testing:

Develop a test plan that includes:

Running the program multiple times with data values that cause all **`true`** blocks to be executed,

AND all **`false`** blocks to be executed.

Check results against the program specifications.

Black Box Testing:

Treat the program like a black box:

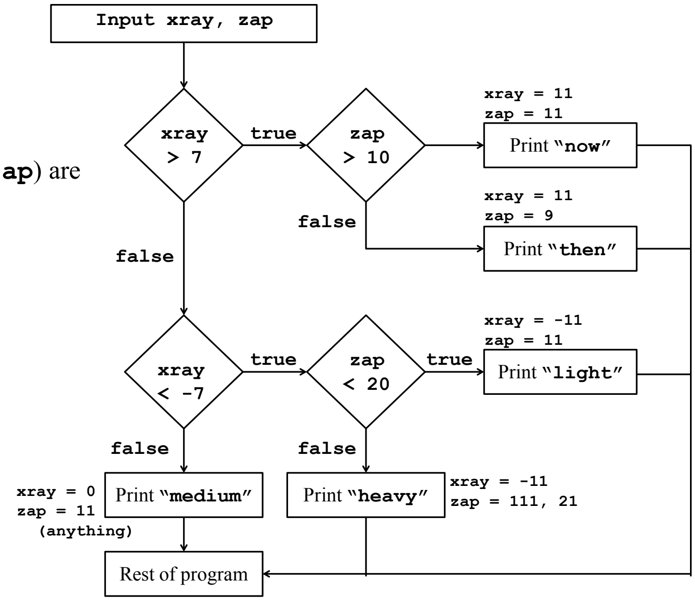
Assume you do not know how the code is written.

Develop test data on program specifications.

Testing Techniques (continued):

Consider the flowchart.

What values of (**`xray`**, **`zap`**) are needed to test each path?


Input xray, zap
xray > 7
true
false
zap > 10
false
xray = 11
zap = 11
Print "now"
xray = 11
zap = 9
Print "then"
xray < -7
true
false
zap < 20
true
false
xray = -11
zap = 11
Print "light"
xray = 0
zap = 11
(anything)
Print "medium"
Print "heavy"
xray = -11
zap = 111, 21
Rest of program