COP 3330, Spring 2013

Control Flow, Intro to Array

Instructor : Arup Ghosh

01-18-13

School of Electrical Engineering and Computer Science
University of Central Florida

Recap - Comparison Operators

- All produce boolean (true or false) (*Difference from C)*
- Equality == (not to be confused with =)
- Inequality !=
- Less than <
- Greater than >
- Less than or equal to <=
- Greater than of equal to >=

Recap - Logical Operators

- **Only between booleans (*Difference from C*)**
- **Logical AND &&**
- **Logical OR ||**
- **Logical NOT ! (Unary)**
- **Example:**
  - `boolean foo = !((a<b) || (c>d));`

Recap - Operator Precedence

Operator	Type	Order of Evaluation

`( )`
`[ ]`
`.`
	Parentheses
	Array Subscript
	Method/Field Access
		Left to right

`++ --`	**Pre-increment/decrement**	**Right to left**

`++ --`
`-`
`!`
	Post-increment/decrement
	Unary minus
	Not
		Right to left

`* / %`	**Multiplicative**	**Left to right**
`+ -`	**Additive**	**Left to right**
`< > <= >=`	**Relational**	**Left to right**
`== !=`	**Equality**	**Left to right**
`&&`	**And**	**Left to right**
`||`	**Or**	**Left to right**
`? :`	**Conditional**	**Right to left**
`= += -= *= /= %=`	**Assignment**	**Right to left**

Recap - Control Flow

- If/else if/else- allows branching of control based on conditions

```
if(a<b)
      System.out.println(“Less”);
else if(a==b)
      System.out.println(“Equal”);
else
      System.out.println(“Greater”);
```

Control Flow

- Switch- Allows branching to multiple pieces of code based on a single variable

```
switch(answer){
        case ‘y’:
            System.out.println(“Confirmed”);
            break;
        case ‘n’:
            System.out.println(“Denied”);
            break;
        default:
            System.out.println(“Not understood”);
}
```

Control Flow

- For- Usually used to loop a certain number of times

```
// Adds up numbers from 1 to n
int sum = 0;
for (int i=1; i<=n; ++i) {
     sum = sum + i;
}
```

Control Flow

- While- Loop as long as a condition is met

```
int n = 1;
while (n<128) {
    System.out.println(n);
    n *= 2;
}
```

Control Flow

- **do-while- Similar to a while loop, except that the loop body is always executed at least once**

```
int n;
do {
        System.out.print(“Enter a number: ”);
        n = stdin.nextInt();
} while (n!=0);
```

Arrays

- Arrays are used for sequential storage of data
- Arrays must be created using the new keyword
  - Example: `int[] a = new int[64];`

Example

- Write a program that reads in 10 numbers then prints them in reverse order.

Arrays

- Arrays can be multi-dimensional
- Example:
  - `int[][] matrix = new int[64][64];`

Example

- Two D Array Example

Assignment #1 Overview

- Read the details carefully
- 3 problems
- Due next Friday