Midterm Review

October 22, 2007

ComS 207: Programming I (in Java)
Iowa State University, FALL 2007
Instructor: Alexander Stoytchev

Midterm 2

- **This Tuesday (Oct 23, 2007)**
- **Location: Curtiss Hall, room 127**
- **Time: 6:30pm – 7:45pm**
- **Try to be there at least 10 minutes early.**
- **If you need extra time you can have it but this is **NOT** a 3 hour exam!**

Midterm Format

- **Covers Sections 1-5 & 7**
- **Also, Searching and Sorting**
- **Format same as Midterm 1**

Midterm Format (last semester)

- **True/False   (10 x 1p each = 10p)**
- **Short answer   (5 x 2p each = 10p)**
- **Code Snippets (5 x 3p each = 15p)**
- **Other Stuff   (3 x 5p each = 15p)**
- **Other Stuff   (3 x 5p each = 15p)**
- **Program 1   (15p)**
- **Program 2   (15p)**
- **Program 3   (15p)**
- **Program 4   (20p)**

- **TOTAL   (130p)**

Midterm Format

- **You don't need to get all 130 points to get an A**
- **100 is a 100**
- **You must get at least 65 points in order to pass this exam**

Midterm Format

- **Drop Deadline is this Friday (Oct 26)**
- **I cannot guarantee that all exams will be graded by then.**
- **If you believe that you did not do well please ask me to grade your exam first at the time when you are submitting it.**

Questions

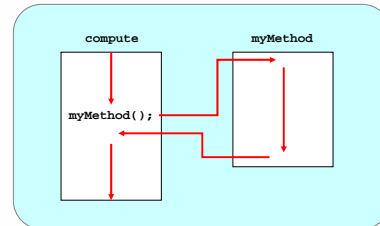## Quick Review of the Material Since Midterm 1

Encapsulation

- An encapsulated object can be thought of as a *black box* -- its inner workings are hidden from the client

- The client invokes the interface methods of the object, which manages the instance data

Client — Methods
Data
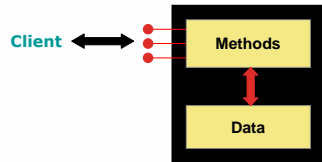
Method Control Flow

- If the called method is in the same class, only the method name is needed

compute   myMethod
myMethod();

Method Control Flow

- The called method is often part of another class or object

main   doIt   helpMe
obj.doIt();   helpMe();
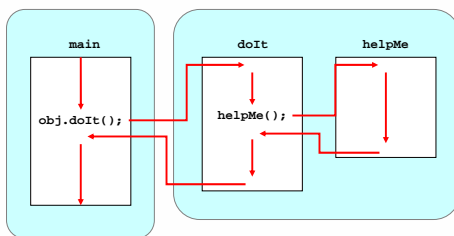
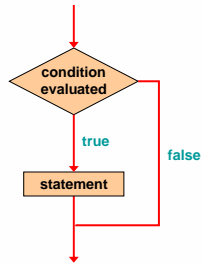## The if Statement

- The *if statement* has the following syntax:

The `condition` must be a boolean expression. It must evaluate to either true or false.

`if` is a Java reserved word

```
if ( condition )
    statement;
```

If the `condition` is true, the `statement` is executed. If it is false, the `statement` is skipped.

2

Logic of an if statement
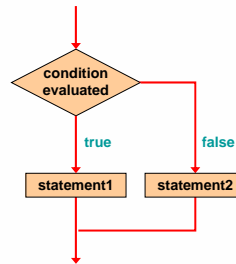
The if-else Statement

- An *else clause* can be added to an `if` statement to make an *if-else statement*

```
if ( condition )
    statement1;
else
    statement2;
```

- If the `condition` is true, `statement1` is executed; if the condition is false, `statement2` is executed

- One or the other will be executed, but not both

Logic of an if-else statement

Logical Operators

- A truth table shows all possible true-false combinations of the terms

- Since `&&` and `||` each have two operands, there are four possible combinations of conditions `a` and `b`

a	b	a && b	a || b
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Boolean Expressions

- Specific expressions can be evaluated using truth tables

total < MAX	found	!found	total < MAX && !found
false	false	true	false
false	true	false	false
true	false	true	true
true	true	false	false

The switch Statement

- The general syntax of a `switch` statement is:

```
switch           switch ( expression )
  and            {
  case              case value1 :
  are                   statement-list1
reserved            case value2 :
  words                 statement-list2
                    case value3 :
                        statement-list3
                    case ...

                 }
```

`switch` and `case` are reserved words

If *expression* matches `value2`, control jumps to here

The switch Statement

- **An example of a switch statement:**

```
switch (option)
{
    case 'A':
        aCount++;
        break;
    case 'B':
        bCount++;
        break;
    case 'C':
        cCount++;
        break;
}
```

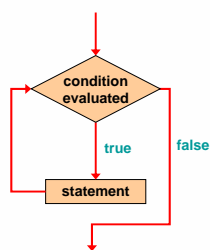## The do Statement

- **An example of a do loop:**

```
int count = 0;
do
{
    count++;
    System.out.println (count);
} while (count < 5);
```

- **The body of a do loop executes at least once**

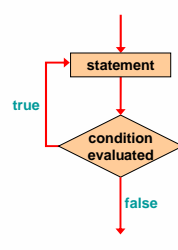- **See ReverseNumber.java (page 244)**

Comparing while and do

The while Loop

The do Loop

The for Statement

- **A *for statement* has the following syntax:**

The *initialization* is executed once before the loop begins

The *statement* is executed until the *condition* becomes false

```
for ( initialization ; condition ; increment )
    statement;
```

The *increment* portion is executed at the end of each iteration
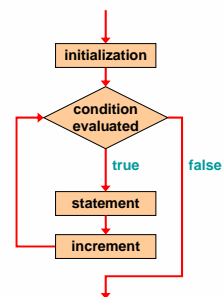
The for Statement

- **An example of a for loop:**

```
for (int count=1; count <= 5; count++)
    System.out.println (count);
```

- **The initialization section can be used to declare a variable**

- **Like a while loop, the condition of a for loop is tested prior to executing the loop body**

- **Therefore, the body of a for loop will execute zero or more times**

Logic of a for loop

The for Statement

- A `for` loop is functionally equivalent to the following `while` loop structure:

```
initialization;
while ( condition )
{
    statement;
    increment;
}
```

Nested Loops

- How many times will the string `"Here"` be printed?

```
count1 = 1;
while (count1 <= 10)
{
    count2 = 1;
    while (count2 <= 20)
    {
        System.out.println ("Here");
        count2++;
    }
    count1++;
}                        10 * 20 = 200
```
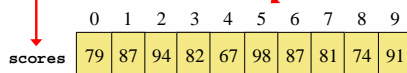
Arrays

- An *array* is an ordered list of values

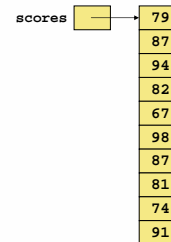The entire array has a single name   Each value has a numeric *index*

```
         0   1   2   3   4   5   6   7   8   9
scores  79  87  94  82  67  98  87  81  74  91
```

An array of size N is indexed from zero to N-1

This array holds 10 values that are indexed from 0 to 9

Arrays

- Another way to depict the `scores` array:

```
scores  →  79
           87
           94
           82
           67
           98
           87
           81
           74
           91
```
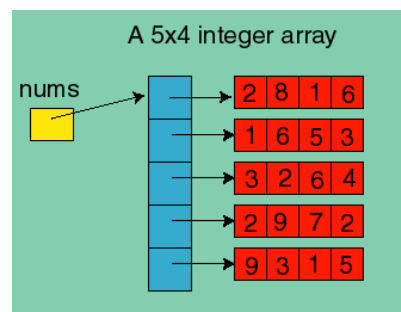
Arrays in Java

- Java represents 2D arrays as an array of arrays!

- In other words, a 2D integer array is really a 1D array of references to 1D integer arrays.

- The concept generalizes to N-dimensions

Anatomy of a 2D Array

A 5x4 integer array

[http://www.willamette.edu/~gorr/classes/cs231/lectures/chapter9/arrays2d.htm]

Example of a regular 2D array
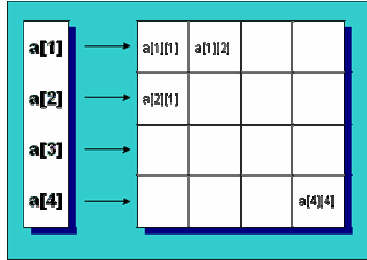
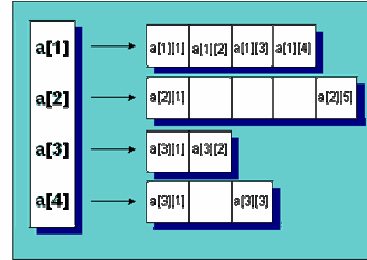| a[1] | → | a[1][1] | a[1]|2] | | |
a[2]	→	a[2][1]			
a[3]	→				
a[4]	→				a[4]|4]

Note: In Java the first index should be 0 not 1!
[http://livedocs.macromedia.com/coldfusion/5.0/Developing_ColdFusion_Applications/arrayStruct2.htm]

Example of a Ragged Array

a[1]	→	a[1][1]	a[1][2]	a[1][3]	a[1][4]
a[2]	→	a[2][1]			a[2][5]
a[3]	→	a[3][1]	a[3][2]		
a[4]	→	a[3][1]		a[3][3]	

Note: In Java the first index should be 0 not 1!
[http://livedocs.macromedia.com/coldfusion/5.0/Developing_ColdFusion_Applications/arrayStruct2.htm]

Searching and Sorting

- **Too much stuff to cover again**

- **Check the slides and programs on the web page**

Animations of Sorting Algoritms

- **http://maven.smith.edu/~thiebaut/java/sort/demo.html**

- **http://www.cs.ubc.ca/spider/harrison/Java/sorting-demo.html**

Sample Homework Solutions

THE END