Midterm Review

October 22, 2007

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.
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.

Method Control Flow

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

The if Statement

- The if statement has the following syntax:

```
if (condition) {
    statement;
}
```

- The condition must be a boolean expression. It must evaluate to either true or false.
- If the condition is true, the statement is executed. If it is false, the statement is skipped.

```java
if (is a Java reserved word) {
    condition;
    statement;
}
```
The if-else Statement

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

```java
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

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| true  | true  |
| false| false| false | true  |
| false| true| false | true  |

The switch Statement

- The general syntax of a switch statement is:

```java
switch ( expression )
{
  case value1 :
    statement-list1
    break;
  case value2 :
    statement-list2
    break;
  case value3 :
    statement-list3
    break;
  default :
    statement-list4
}
```

- 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

\[ \begin{array}{|c|c|}
\hline
\text{The while Loop} & \text{The do Loop} \\
\hline
\text{condition} & \text{statement} \\
\text{evaluated} & \text{true} \\
\text{false} & \text{false} \\
\hline\end{array} \]

The for Statement

• A for statement has the following syntax:

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

• 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

\[ \begin{array}{|c|}
\hline
\text{Initialization} \\
\hline
\text{condition} \\
\text{evaluated} \\
\text{false} \\
\hline
\text{statement} \\
\hline
\text{Increment} \\
\text{false} \\
\hline\end{array} \]
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?

```java
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

<table>
<thead>
<tr>
<th>scores</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>79</td>
<td>87</td>
<td>94</td>
<td>82</td>
<td>67</td>
<td>98</td>
<td>87</td>
<td>81</td>
<td>74</td>
<td>91</td>
</tr>
</tbody>
</table>

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 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

```
nums
-2 8 1 6
-1 6 5 3
-3 2 6 4
-2 9 7 2
-9 3 1 5
```
Example of a regular 2D array

Note: In Java the first index should be 0 not 1!

Example of a Ragged Array

Note: In Java the first index should be 0 not 1!

Searching and Sorting
- Too much stuff to cover again
- Check the slides and programs on the web page

Animations of Sorting Algorithms
- http://maven.smith.edu/~thiebaut/java/sort/demo.html

Sample Homework Solutions

THE END