Selection Statements &
Exam 1 Review
Recitation – 2/13/2009

CS 180
Department of Computer Science,
Purdue University
Announcements

- **EXAM**
  - Feb 18, Wednesday
  - 6:30-7:30pm
  - MTHW 210
  - Chapter 0 to 5
  - 20 Multiple choice, 3 programming

- **Project 4 Posted**
  - Milestone due Feb 18, 10pm
    - This will be graded! Make sure you submit
  - Final due Feb 25, 10pm
Syntax for the if Statement

if ( <Boolean expression> )
  <then block>
else
  <else block>

Boolean Expression Evaluated to either true or false

If ( testScore < 70 )
  JOptionPane.showMessageDialog(null, "You did not pass");
else
  JOptionPane.showMessageDialog(null, "You passed");
Relational Operators

<  // less than
<=  // less than or equal to
==  // equal to
!=  // not equal to
>  // greater than
>=  // greater than or equal to

testScore < 80

testScore * 2 >= 350

30 < w / (h * h)

x + y != 2 * (a + b)

2 * Math.PI * radius <= 359.99
Compound Statements

Use braces if the <then> or <else> block has multiple statements.

```java
if (testScore < 70) {
    JOptionPane.showMessageDialog(null, "You did not pass");
    JOptionPane.showMessageDialog(null, "Try harder next time");
} else {
    JOptionPane.showMessageDialog(null, "You did pass");
    JOptionPane.showMessageDialog(null, "Keep up the good work");
}
```
Control Flow with no else

```java
if (testScore >= 95)
    JOptionPane.showMessageDialog(null, "You are an honor student");
```

if (testScore >= 95)
    JOptionPane.showMessageDialog(null, "You are an honor student");
The Nested-if Statement

The then and else block of an if statement can contain any valid statements, including other if statements. An if statement containing another if statement is called a nested-if statement.

```java
if (testScore >= 70) {
    if (studentAge < 10) {
        System.out.println("You did a great job ");
    } else {
        System.out.println("You passed"); //test score >= 70
    }
    //and age >= 10
} else { //test score < 70
    System.out.println("You did not pass");
}
```
if (score >= 90)
    System.out.print("Your grade is A");

else if (score >= 80)
    System.out.print("Your grade is B");

else if (score >= 70)
    System.out.print("Your grade is C");

else if (score >= 60)
    System.out.print("Your grade is D");

else
    System.out.print("Your grade is F");
Matching else

Are A and B different?

A

```java
if (x < y)
    if (x < z)
        System.out.print("Hello");
else
    System.out.print("Good bye");
```

B

```java
if (x < y)
    if (x < z)
        System.out.print("Hello");
else
    System.out.print("Good bye");
```

Both A and B mean...

```java
if (x < y) {
    if (x < z) {
        System.out.print("Hello");
    } else {
        System.out.print("Good bye");
    }
}
```

Each else paired with nearest unmatched if -- use braces to change this as needed.
Boolean Operators

&& // AND
|| // OR
!  // Not

if (temp >= 65 && temp <= 80)
    System.out.println("Let's walk");
else
    System.out.println("Let's drive");

if (65 <= temp <= 80)
    System.out.println("Let's walk");
else
    System.out.println("Let's drive");
Semantics of Boolean Operators

- Truth table for boolean operators:

| p   | q   | p && q | p || q | !p |
|-----|-----|--------|--------|----|
| false | false | false | false  | true |
| false | true  | false | true   | true |
| true  | false | false | true   | false |
| true  | true  | true  | true   | false |

- Sometimes true and false are represented by 1 and 0 (NOT in Java).
- In C and C++, 0 is **false**, everything else is **true**.
Short-Circuit Evaluation

- Stop the evaluation once the result of the whole expression is known
- Example:
  - Let $p = 1$, $q = 2$, $r = 3$, $s = 4$

```
\begin{align*}
  & p > q \lor s > r \\
  & s > r \lor p > q \\
  & p < q \land r < s \\
  & s < r \land p < q
\end{align*}
```
Simple switch statement

```java
switch ( N ) {
    case 1: x = 10;
    case 2: x = 20;
    case 3: x = 30;
}
```

Must be constant!
switch with break, and default

```java
switch (N) {
    case 1: x = 10;
        break;
    case 2: x = 20;
    case 3: x = 30;
        break;
    default: x = 0;
}
```
Missing case body

```java
switch ( N ) {
    case 1: x = 10;
        break;
    case 2:
    case 3: x = 30;
        break;
    default: x = 0;
}
```
Write a program that print “A” if the input (x) is 10, 20, or 30. Print “B” if the input is 40, print C if otherwise.

```java
if(x == 10 || x == 20 || x == 30)
    System.out.println("A");
else if(x == 40)
    System.out.println("B");
else
    System.out.println("C");
```

```java
Switch(x) {
    case 10: 
        System.out.println("A");
        break;
    case 20: 
        break;
    case 30: 
        System.out.println("A");
        break;
    case 40: 
        System.out.println("B");
        break;
    default:
        System.out.println("C");
}
```
String a = “Hello World”;
String b = new String(“Hello World”);
a.subString(1, 5) \(\Rightarrow\) “ello” (Extract from i to j-1)
a.indexOf(“Wor”) \(\Rightarrow\) 6 (Zero-based index)
a.indexOf(“Ello”) \(\Rightarrow\) -1 (Case sensitive)
a.length() \(\Rightarrow\) 11
a.subString(1, 5) + a.subString(6, 9) \(\Rightarrow\) “elloWor”
a.subString(1, 5) + 8 + 9 \(\Rightarrow\) “ello89”
a.subString(1, 5) + (8 + 9) \(\Rightarrow\) “ello17”
System.out.println(“Study for Exam”);
System.out.print(“Study for Exam”);
JOptionPane.showMessageDialog(null, “Studying”);

Scanner s = new Scanner(System.in);
s.next(); ➔ Read the next word
JOptionPane.showMessageDialog(null, “Your name?”);
Primitive Types

- int, long, double, float, ...
- Do not need to call `new` to allocate space
- Can assign a value after declaration
- Example: `int x = 5;`

Reference Data Types

- String, Maze, Calculator, ...
- The variable refers to the object allocated by `new` operator
- Must be created using `new` before manipulation otherwise it is `null`
- Example: `Maze m = new Maze("Name");`
# Facts about Primitive Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Content</th>
<th>Default Value</th>
<th>Size (bytes)</th>
<th>Minimum Value</th>
<th>Maximum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte</td>
<td>Integer</td>
<td>0</td>
<td>1</td>
<td>-128</td>
<td>127</td>
</tr>
<tr>
<td>short</td>
<td>Integer</td>
<td>0</td>
<td>2</td>
<td>-32768</td>
<td>32767</td>
</tr>
<tr>
<td>int</td>
<td>Integer</td>
<td>0</td>
<td>4</td>
<td>-2147483648</td>
<td>2147483647</td>
</tr>
<tr>
<td>long</td>
<td>Integer</td>
<td>0</td>
<td>8</td>
<td>-9.22337E+18</td>
<td>9.22337E+18</td>
</tr>
<tr>
<td>float</td>
<td>Real</td>
<td>0.0</td>
<td>4</td>
<td>-3.40282347 x 10^{38}</td>
<td>3.40282347 x 10^{38}</td>
</tr>
<tr>
<td>double</td>
<td>Real</td>
<td>0.0</td>
<td>8</td>
<td>-1.7977 x 10^{308}</td>
<td>1.7977 x 10^{308}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order</th>
<th>Group</th>
<th>Operator</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Subexpression</td>
<td>( )</td>
<td>Starting with innermost ()</td>
</tr>
<tr>
<td></td>
<td>Unary operators</td>
<td>-, +</td>
<td>Left to right.</td>
</tr>
<tr>
<td></td>
<td>Multiplicative operators</td>
<td>*, /, %</td>
<td>Left to right.</td>
</tr>
<tr>
<td></td>
<td>Additive operators</td>
<td>+, -</td>
<td>Left to right.</td>
</tr>
</tbody>
</table>
Generating Random Numbers

- Generate a random **integer** between A & B
  \[(\text{int})(\text{Math.random}() \times (B - A + 1)) + A\]

- [0, 1)

- [0, B-A+1)

- [0, B-A]

- [A, B]
Generating Random Numbers

- Generate a random \textbf{real number} between A & B

\[
\text{Math.random()} \times (B - A) + A
\]

- [0, 1)
- [0, B-A)
- [A, B)
Conventional Class Definition Structure

class
{

Import Statements
Class Comment
Class Name
Data Members
Methods (incl. Constructor)
}
import java.util.Date;

/**
 * Book -- a book that knows only its name
 * @author Henry David Thoreau
 **/

class Book {
    private String name;
    private Date dateMade;

    public Book() {
        name = "I have no name";
        dateMade = new Date();
    }

    public String getName() {
        return name;
    }

    public void setName(String newName) {
        name = newName;
    }
}
Write a function to compute the probability of a student getting a $x$ points in the exam. The input parameters are $x$, $\mu$ (class average), $\sigma$ (variance). The probability function is given by

$$p(x, \mu, \sigma) = k e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Let $k = \frac{1}{\sigma\sqrt{2\pi}}$

```java
define the probability function
public double Probability(double x, double mu, double sigma)
{
    double k = 1.0 / sigma / Math.sqrt(2 * Math.PI);
    return k * Math.exp((-(x-mu)*(x-mu)) / 2 / sigma / sigma);
}
```
Quiz

- Java is a machine independent language. True or false?
- Any switch statement can be written using if statements. True or false?