Recitation 1
Java and Eclipse
Data Types, Variables, Logical Operators

February 13, 2012
Outline

• Administrative
• Java and Eclipse
• Data Types
• Variables
• Logical Operators
• Homework 1
Reminders

Office Hours
- Wednesday 5pm - 10pm
- Thursday 5pm - 10pm

2 Friday Quizzes: March 9 & April 13
Review session before all quizzes and finals.
- Wed. March 7 7pm - 9pm
- Wed. April 11 7pm – 9pm
- Wed. May 16 7pm – 9pm

Academic Honesty Form – Read it! Sign it!
Grading

Homework (10)  40%
Active Learning Exercises  10%
Quiz 1 (March 9)  12%
Quiz 2 (April 13)  12%
Final Exam (TBA)  20%
Recitation Participation  6%
## Schedule

<table>
<thead>
<tr>
<th>February</th>
<th>March</th>
<th>April</th>
<th>May</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction to Java:</strong> Operators, Control, Data Types, Methods, Classes &amp; Objects, Arrays &amp; ArrayLists</td>
<td><strong>Inheritance, Interfaces</strong></td>
<td><strong>Streams, Sensors &amp; Threads</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Graphical User Interfaces</strong></td>
<td><strong>Quiz 1</strong></td>
<td><strong>Quiz 2</strong></td>
<td><strong>Data Structures</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Numerical Methods</strong></td>
<td></td>
<td><strong>Final</strong></td>
</tr>
</tbody>
</table>
Homework

• Hard copy available in lecture a week before due date (2 weeks if quiz)
• Electronic copy available a week before the hard copy
• Due on Friday at 12 noon.
• Submissions accepted until 12 noon on Monday, with a 30-point penalty
• 1 no-penalty late submission (still before Monday 12 noon) – You still must turn in your pset!
• A submission is considered late if it has a late tag on the website
• Make sure you submit your .java files, and not your .class files
• Group multiple files in a .zip folder
• Every .java file must start with your name, MIT email and section number
• We do not omit your lowest problem set.
Active Learning Exercises

• Java files to be downloaded before almost every lecture
• Exercises are included in the lecture notes and often use the downloaded files
• Submit your solutions in the Homework section of the website before 8pm. No late submissions allowed.
• Java solutions are released in the Homework section
• PDF solutions and lecture notes including the solutions are released in the In-Class Exercises section
• You can download the lecture notes and the Java files and submit your exercises a week before lecture
• Complete the exercises for 30 lectures to get full credit (10% of final grade)
What you Installed

Eclipse

Java
JRE / JDK
Multiple Virtual Machines (JVM)

Hardware

One Compiler

Compiler

Text Editor

.java File

.class File

Intel 32 bit JVM

Intel 64 bit JVM

AMD 64 bit JVM

Intel 32 bit

Intel 64 bit

AMD 64 bit
public class MileConvert {
    public static void main(String[] args) {
        double circum = 24859.82 * 5280; // earth circumference in feet
        int minutesInCircle = 360 * 60;
        double nautMile = circum / minutesInCircle;
        System.out.println("1 Nautical Mile = " + nautMile + " feet");
    }
}
Don't submit your .class files!
# Java Data Types

8 *primitive* data types

<table>
<thead>
<tr>
<th>Type</th>
<th>Size (bits)</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>1</td>
<td>true or false</td>
</tr>
<tr>
<td>char</td>
<td>16</td>
<td>ISO Unicode character set</td>
</tr>
<tr>
<td>byte</td>
<td>8</td>
<td>-128 to 127</td>
</tr>
<tr>
<td>short</td>
<td>16</td>
<td>-32,768 to 32,767</td>
</tr>
<tr>
<td>int</td>
<td>32</td>
<td>-2,147,483,648 to 2,147,483,647</td>
</tr>
<tr>
<td>long</td>
<td>64</td>
<td>-9,223,372,036,854,775,808L to 9,223,372,036,854,775,807L</td>
</tr>
<tr>
<td>float</td>
<td>32</td>
<td>+/- 3.4E+38F (6-7 significant digits)</td>
</tr>
<tr>
<td>double</td>
<td>64</td>
<td>+/- 1.8E+308 (15 significant digits)</td>
</tr>
</tbody>
</table>
Java Data Types

Which data type would you use for:

```java
int studentCount = 142;
char firstLetter = 'a';
float weight = 180.6F;
double area = Math.PI * 5.0 * 5.0;
boolean enjoy100 = true;
long theNumberOne = 1L;
double largeNumber = 1.0E100;
```
In practice, we will mostly use:

- boolean to represent logic
- int, long and double to represent numbers

For text, we will use Strings, which are chains of char. e.g. String text = "Welcome to 1.00";

A String is an object, not a primitive data type.
Variables

Data is held in variables

```
int studentCount = 142;
```

- The value on the right is assigned into the left variable name.
- The type of each variable must be declared: Java is a strongly-typed language.
- Variable names typically start with a lowercase letter.
- The variable value must "fit" in the variable type.
Are these variable declarations acceptable? If yes, are they ideal?

```java
boolean b = 1;

double studentCount = 142;

byte preRegCount = 110;

int 2 = facultyCount;
```
Branching: if ... else

• if ... else

```java
if (x == 0)
    System.out.println("x is Zero");
else
    System.out.println("x is NonZero");
```

• if ... else if ... else

```java
if (x < 0)
    System.out.println("x is Negative");
else if (x == 0)
    System.out.println("x is Zero");
else
    System.out.println("x is Positive");
```
Branching: if ... else

• The **else** statement is not required to terminate branching.
  
  ```java
  // e.g. Take the absolute value of x
  if (x < 0)
    x = -x;
  ```

• Use braces `{ }` to execute multiple statements.
  
  ```java
  // e.g. Take the absolute value and notify the user
  if (x < 0){
    x = -x;
    System.out.println("x has been converted");
  }
  ```
Iteration (Loops)

```plaintext
while (condition to continue)
{
    // repeat as long as condition = true
}
```

```plaintext
do
{
    // run once and repeat as long as condition = true
}
while (condition to continue)
```

```plaintext
for (initial statement; condition to continue; increment statement)
{
    // execute initial statement
    // repeat as long as condition = true
    // execute increment statement after each repetition
}
```
Homework 1
Magnetic Inductance
Due: February 17, 2012

Compute magnetic inductance for 3 different types of antennae:

1. Line antenna
2. Coil antenna
3. Rectangular antenna

Also, if (hint) the user selects a coil antenna, and or a rectangular antenna, you will calculate the mutual inductance.

Considerations:
• Ask the user which antenna type
• Parse user inputs using input dialog
• Execute the user’s desired calculations
• Print inductance value(s)
• Depending on antenna type, you will also be calculating mutual inductance

Image by MIT OpenCourseWare. Adapted from Figure 4.8 Finkenzeller, Klaus (2003). RFID Handbook (2nd Edition). Wiley.