

### Introduction to Java

#### Objectives:

1. To set up your SUN account to print with a2ps.
2. To set up your SUN account to compile and run Java applications.
3. To practice writing simple Java applications.

**Preparation:** Before the exercise read the following chapters in *Java: How to Program* by Deitel and Deitel, fourth, fifth or sixth edition: Preface, Chapters 1, 2, 3, 4, 5 and 6. Since these chapters are written for the beginning programmer, many sections can be skimmed. We recommend that you study the programs and if you can explain each line then skip over the text. Otherwise, read enough to understand the programs.

#### Assignment:

1. **Set up Printing:** If you have not set up your account to use a2ps, do so by adding the following lines of code to your `.cshrc` file in your home directory.

```
#####
# Room 213 Dana is old room number 231 Dana
setenv PRINTER dana213-lp1

# The next "alias" line sets an alias "print" to print 2 pages/sheet
# using a2ps with no header page.
# Use by typing after the Unix prompt "%" :
#   % print filename

alias print 'a2ps  "\!*" -o - | lp -o nb -d $PRINTER'

# The next "alias" line changes your default printer.
# For printer in 213 Dana, type
#   % chpr  dana213-lp1
# For printer in 164 Breakiron, type
#   % chpr  brki164-lp1
# For printer in 167 Breakiron, type
#   % chpr  brki167-lp1

alias chpr 'setenv LPDEST "\!*" ; setenv PRINTER "\!*" '
#####
```

After you have added the code and saved the `.cshrc` file, do `source ~/.cshrc` in the Unix shell window where you will be printing. If a file has a `.java` extension, a2ps assumes it is a Java file and bolds all reserved words. All Java files handed in for this course must be printed with a2ps.

**2. Set up Java 2:** If your SUN account is not set up to compile and run Java applications, do the following:

1. Create a new directory `JavaClasses` under your home directory.
2. Add the following two lines to the `.cshrc` file in your home directory to set the environment variable `CLASSPATH`.

```
### CLASSPATH - environment variable for JAVA
setenv CLASSPATH ./home/accounts/.../username/JavaClasses
```

using the **full path name** of your directory where the java compiler is to look for java classes, e.g., `JavaClasses`. If you add new paths, separate the paths with `":"`.

3. We will be using Java 2 in this course. To use Java 2 (version 1.4), add to the path variable BEFORE `/bin` in your `.cshrc` file the following:

```
/usr/local/jdk-1.4/bin
```

4. Do the following in a Unix shell window to check for errors and activate the new changes in your `.cshrc` file:

```
% source ~/.cshrc
```

**3. Compile and Run a Simple java Application:** Using emacs editor, type in the following Java application program. The java code should be colorized by emacs. If not seek help.

```
// A first program in Java

public class Welcome {

    public static void main ( String args []) {

        System.out.println( "Welcome to Java Programming!" );
    }
}
```

Save the file as **Welcome.java**. Note the file name **MUST** have the same name as the class in the file plus the `.java` extension!

For the above application, the class name is "Welcome".

To compile a java application, you type

```
% javac Welcome.java
```

which creates a file of Java byte code called `Welcome.class`.

To run a java application, you use the classname not the file name! In this case,

```
% java Welcome
```

To check which version of Java you are using, type

```
% java -version
```

If you have are having trouble compiling and running the above program, ask for help!

- 4. Using Mercer's Keyboard Class** Input and output in Java can be a bit tricky for beginners. A simple approach is to use a third-party class, i.e., a non-Sun class, for input and the `System.out.println()` method for output. We will use the `TextReader` class from Rick Mercer's textbook *Computing Fundamentals with Java*.

Copy the file `~cs355/public_html/S2004/TextReader.java` to your `JavaClasses` directory.

To use, you create a `keyboard` object as follows:

```
TextReader keyboard = new TextReader();
```

then to read an integer you call the `readInt()` method such as

```
x = keyboard.readInt();
```

Look at the code to see what other methods are available. To use this class, the file needs to be in the same directory as your Java program or in your `CLASSPATH`, e.g., your `JavaClasses` directory. You do *not* need nor want to `import` the `TextReader` class into your program.

Write a Java application to input three floating point numbers as doubles using the `TextReader` class and print out their sum using `System.out.println()` method for output.

- 5. Using JOptionPane from Swing API:** The Java 2 *Swing Application Programming Interface* (API) provides us with a second easy way to do input and output. The class **JOptionPane** provides simple dialog boxes for input and output much like Visual Basic.

Write a Java application to input three floating point numbers as doubles and print out their sum. Use the **JOptionPane** class in the **Swing** API for input. See example on page 69 in fourth edition (47 in fifth, 104-107 in sixth) of Java text. Print out the answer using `System.out.println()` method for output.

- 6. Formatting Output for doubles:** Write a Java application to input ten floating point numbers as doubles and print out their average with two decimal places. To control the outputting of doubles, see page 384 in fourth edition (142 in fifth, 102 in sixth explains the new `printf` way) of Java text. Print out the answer using `System.out.println()` method for output.

- 7. Read "Java for C++ Programmers":** Read carefully, the tutorial by Marvin Solomon at

<http://www.cs.wisc.edu/~solomon/cs537/java-tutorial.html>

Skip the section on **Threads** for now.

If you have trouble accessing Marvin Solomon's web page, a local Bucknell copy is available on CS475 web page.

- 8. Reading from a file:** Write a Java application to read the lines in a text file and print them to the shell window using the `System.out.println()` method.

Hint: You will find the information needed by a careful reading of Solomon's web pages.

**9. Explore the CS475 Java Resources Web Page:** Spend some time exploring the CSCI 475 Java Resources web page at

`http://www.eg.bucknell.edu/~cs475/F04-S05/javaResources.html`

Since we will be using it later, we recommend you bookmark it.

**Hand In:** For Exercises 4, 5, 6 and 8, combine all the Java listings and outputs from runs into one handin file with a **.java** extension. Print using the **print** alias set up in Exercise 1.