

CS 111 Summary

User Environment

Terminal window: Prompts user for commands.

BASH shell tips:

- <tab> for command line completion.
- <up>/<down> to step backward/forward through command history.
- !<start of command> will re-execute most recent command matching start
- <left>/<right>/<Ctrl-a>/<Ctrl-e> to navigate and edit line

Common UNIX commands:

- man <command> – get help for a command
- apropos <keyword> – get help for a keyword
- ls – list current directory contents
- cd <directory> – change directory
- mkdir <directory> – make directory
- rmdir <directory> – remove directory
- more/less <file> – show contents of text file
- mv <file> <new file location/name> – move/rename file/directory
- cp <file> <new file location/name> – copy file
- rm <file> – remove file
- yppasswd – change password

Common applications (append “<space>&” in BASH shell for long running applications)

- firefox – web browser
- eclipse – Java integrated development environment (IDE)
- java / javac – Java virtual machine / compiler

Java Application Process

1. Edit a plain-text source code file <classname>.java using a text editor (e.g. gedit, emacs, vim, notepad, text edit, etc.) or an integrated development environment (IDE).
2. Compile (translate) source code <classname>.java into Java bytecode class file <classname>.class by entering the command: javac <classname>.java
3. Interpret (translate and execute) Java bytecode class file with the Java Virtual Machine (JVM) interpreter by entering the command: java <classname>

A Simple Java Application

In file Hello.java:

```
public class Hello {  
  
    public static void main(String[] args) {  
  
        // insert program code here, e.g.:  
        System.out.println("Hello, world!");  
  
    }  
  
}
```

Note: The class name after `public class` has to match the filename before the “.java” extension.

Printing in Java

- To print a line to standard output: `System.out.println(something to print);`
- A String is a string/sequence of characters. We can create a Java String by enclosing characters between double quotes, e.g. `"Hello, world!"`.
- The backslash (`\`) is the *escape character* for Java Strings. Anything after the backslash is given special meaning in a String.
- A double-quote may be included in a String by placing a backslash (`\`) before it.
- A backslash may be included in a String by using two backslashes.
- There are other useful backslash sequences as well, e.g. newline (`\n`) and tab (`\t`).
- To print to standard output without starting a new line at the end, use `System.out.print(something to print);`
- To print a formatted String, use `System.out.printf(format String, value1, ..., valueN);`
- Format Strings contain formatting placeholders (e.g. `%s` for String, `%d` for integer, `%f` for floating point number) that have print representations of following values substituted in place.