

## IntelliJ Terminal Commands Worksheet

1. Note - the terms *terminal* and *shell* are used interchangeably.
2. Unzip the project and open the project in IntelliJ:
  - `Average.java` - reads a sequence of integers from stdin, computes their average, and writes the average to stdout
  - `RangeFilter.java` - reads a sequence of integers from standard input and writes those values between `lo` and `hi` to stdout
  - `numbers.txt` - contains 1000 integers
  - `presidents.txt` - contains the names and terms of US presidents
  - `top-500.txt` - contains Rolling Stone's 500 greatest songs of all time
3. Lift → Terminal
4. The `>` is called the terminal or shell *prompt*.
  - Change the prompt by entering the command:
    - `PS1="> "`
5. TAB completion. Use the TAB key to complete command names and file names.
6. History. Press the up arrow key (repeatedly) to see the previous command(s). You can edit and re-run them using the right and left arrow keys to position the cursor and the delete / backspace key to edit the command line.
7. The shell has many commands, e.g., `javac-introcs program.java` and `java-introcs program` to compile and execute Java programs, respectively. Some other useful shell command you should try:

Shell command	Description
<code>&gt; ls</code>	List the contents of the current directory
<code>&gt; ls -l</code>	A verbose list of the contents of the current directory
<code>&gt; pwd</code>	Print the name of the current directory
<code>&gt; rm filename</code>	Remove / delete the file named filename; your shell may ask you to confirm the file removal - type <code>y</code> to confirm; <code>n</code> to cancel
<code>&gt; less filename</code>	Print a file, page by page. Use the spacebar to page; use <code>q</code> to quit.
<code>&gt; sort filename</code>	Sort the lines in a file
<code>&gt; uniq filename</code>	Print the unique lines in a file
<code>&gt; grep pattern filename</code>	Find occurrences of a pattern in a file, e.g., <code>grep John presidents.txt</code>

8. Compile `Average.java` and `RangeFilter.java`

9. Run the Average program

```
> java-introcs Average
10.0  5.0  6.0
3.0   7.0 32.0
```

Use<ctrl-d> (Mac) or <ctrl-z><enter> (Windows) to signal the end-of-file.

Answer:

10. Run the RangeFilter program

```
> java-introcs RangeFilter 4 7
1 20 30 2 4 6 -1
100 6 7
34 0
```

Use<ctrl-d> (Mac) or <ctrl-z><enter> (Windows) to signal the end-of-file.

Answer:

11. Redirect stdin: Run the Average program redirecting numbers.txt to stdin:

```
> java-introcs Average < numbers.txt
```

Answer:

12. Redirect stdin: Run the RangeFilter program redirecting numbers.txt to stdin:

```
> java-introcs RangeFilter 1 5 < numbers.txt
```

Answer:

13. Redirect stdin and stdout: Run the Average program redirecting numbers.txt to stdin and stdout to the file averages.txt

```
> java-introcs Average < numbers.txt > averages.txt
```

Answer:

14. Remove the file averages.txt

```
> ls
> rm averages.txt
> ls
```

15. Redirect stdin: Run the RangeFilter program redirecting `numbers.txt` to stdin and stdout to the file `results.txt`

```
> java-introcs RangeFilter 10 20 < numbers.txt > results.txt
```

Answer:

16. Redirect stdin and stdout: Run the RangeFilter program redirecting `numbers.txt` to stdin and redirecting stdout to `ranges.txt`. Then run the Average program redirecting `ranges.txt` to stdin

```
> java-introcs RangeFilter 10 20 < numbers.txt > ranges.txt
> java-introcs Average < ranges.txt
```

Answer:

17. Redirect stdin and piping: Run the RangeFilter program redirecting `numbers.txt` to stdin and piping stdout to the Average program

```
> java-introcs RangeFilter 10 20 < numbers.txt | java-introcs Average
```

Answer:

18. Use the `less` command to print the `presidents.txt` and `top-500.txt` files to stdout. Recall - use the spacebar to page.

```
> less presidents.txt
> less top-500.txt
```

19. Sort the `presidents.txt` file

Answer:

20. Sort the `presidents.txt` file redirecting stdout to a `sorted-pres.txt`

Answer:

21. Use the `less` command to view the `sorted-pres.txt`

Answer:

22. Sort the `presidents.txt` file and pipe stdout to the `less` command

Answer:

23. What songs do the Beatles have in the top 500?

Answer:

**Answers:**

**9. Average is 10.5**

**10.**

**4**

**6**

**6**

**7**

**11. Average is 50.117**

**12. Many values between 1 and 5**

**13. Examine file averages.txt: Average is 50.117**

**15. Examine file results.txt. Many values between 10 and 20**

**16. Average is 15.315315315315315**

**19. sort presidents.txt**

**20. sort presidents.txt > sorted-pres.txt**

**21. less sorted-pres.txt**

**22. sort presidents.txt | less**

**23. grep Beatles top-500.txt**

## IntelliJ Terminal Commands

- Lift → Terminal
- The `>` is called the terminal or shell *prompt*
  - Change the prompt entering the command: `PS1="> "`
- TAB completion. Use the TAB key to complete command names and file names
- History. Press the up arrow key (repeatedly) to see the previous command(s). You can edit and re-run them using the right and left arrow keys to position the cursor and the delete / backspace key to edit the command line.

Command	Description
<code>pwd</code>	current working directory
<code>ls</code> <code>ls -l</code>	list directory
<code>rm filename</code>	remove/delete filename (if prompted - use <code>y</code> to confirm; <code>n</code> to cancel)
<code>javac-introcs MyClass.java</code>	compile MyClass.java
<code>java-introcs MyClass arg0 arg1 arg2...</code>	run program MyClass with command line arguments
<code>java-introcs MyClass arg0 arg1 ... &lt; infile</code>	Redirect standard input from file named <i>infile</i>
<code>java-introcs MyClass arg0 arg1 ... &gt; outfile</code>	Redirect standard output to file named <i>outfile</i>
<code>java-introcs MyClass arg0 arg1 ... &lt; infile &gt; outfile</code>	Redirect standard input from file named <i>infile</i> and redirect standard output to file named <i>outfile</i>
<code>java-introcs Program1   java-introcs Program2</code>	pipng
<code>java-introcs Program1 &lt; input.txt   more</code> <code>java-introcs Program1 &lt; input.txt &gt; output.txt</code> <code>java-introcs Program1 arg1   java-introcs Program2 arg2</code>	combinations
<b>ctrl-c</b>	terminate a program
<b>Mac: enter control-d    Windows: enter cntl-z enter</b>	end of file
<code>jshell-introcs</code>	start jshell - use ctrl-d to end
<code>less, uniq, grep, sort, cat, head, tail</code>	Terminal (shell) commands

