

The following examples demonstrate how to perform command line processing, similar to the examples shown in class:

Example #1:

```
public class simpleCommandLineProcessing
{
    public static void main(String[] args)
    {
        for (int i = 0; i < args.length; i++)
            System.out.println(args[i]);
    }
}
```

Sample run:

```
% java simpleCommandLineProcessing arg1 arg2 56 -arg4 99arg5
arg1
arg2
56
-arg4
99arg5
```

Example #2:

```
/* This program takes one of two command line arguments:

The first argument is a switch indicating whether the second argument is to be
printed out in upper, or lower, case.

The second argument is a word that is printed out to the screen in accordance with
the first argument, namely all upper case, or all lower case.

The following is how this program is expected to be used:

    java sclp -u|-l anyWord

Notes:
- Command line arguments are separated by spaces
- Command line arguments starting with a dash must be lower case
- Command line arguments starting with a dash, which are sometimes optional,
  are called "switches."
- The expression "-u|-l" means that the either a -u or a -l must be specified
*/
public class sclp
{
    public static void main(String[] args)
    {
        if (args.length != 2)
        {
            System.out.println("usage: java sclp -u | -l anyWord");
            System.exit(-1);
        }

        if (args[0].equals("-u"))
            System.out.println(args[1].toUpperCase());
        else if (args[0].equals("-l"))
            System.out.println(args[1].toLowerCase());
        else
        {
            System.out.println("The command line switch must be a '-u' or a '-l'");
            System.out.println("usage: java sclp -u | -l anyWord");
        }
    }
}
```

Sample runs:

```
% java sclp -u hello
HELLO

% java sclp -l hELLo
hello

% java sclp hello
usage: java sclp -u | -l anyWord

% java sclp -d hello
The command line switch must be a '-u' or a '-l'
usage: java sclp -u | -l anyWord
```

Example 3: We will return to this section later in the course

```
import java.io.BufferedReader;
import java.io.InputStreamReader;

import java.io.FileReader;

import java.io.IOException;
import java.io.EOFException;

/* This program takes one required and one optional command line argument:

The first argument is an optional switch indicating whether the filename is to be
printed out in upper, or lower case.

The second argument is is required; this argument specifies the name of a file to
be printed to the terminal/terminal window. The contents of this file are either printed out
unchanged, uppercase or lower case depending upon whether the first, optional argument
is specified.

The following is how this program is expected to be used:

    java type [-u|-l] filename

Notes:
- Command line arguments are separated by spaces
- Command line arguments starting with a dash must be lower case
- Command line arguments starting with a dash, which are sometimes optional,
are called "switches."
- The expression [-u|-l] means that the arguments are optional and that either a
-u, a -l, or no option is permitted; however, a -u and a -l cannot both be used together
*/

public class type
{
    public static void main(String[] args)
    {
        WhichCase theCase = WhichCase.unchanged; // unchanged => output file contents w/o changing the case
        FileReader fr = null;

        if (args.length < 1 || args.length > 2)
        {
            System.out.println("usage: java sc1p [-u | -l] filename");
            System.exit(-1);
        }

        if (args[0].equals("-u") && args.length == 2)
            theCase = WhichCase.upper;
        else if (args[0].equals("-l") && args.length == 2)
            theCase = WhichCase.lower;
        else if (args[0].charAt(0) == '-' || args.length == 2 && args[0].charAt(0) != '-')
        {
            System.out.println("usage: java type [-u | -l] filename");
            System.exit(-1);
        }

        try
        {
            if (args.length == 1)
                fr = new FileReader(args[0]);
            else
                fr = new FileReader(args[1]);

            BufferedReader fd = new BufferedReader(fr);

            while (true)
            {
                String line = fd.readLine();

                if (line == null)
                    break;

                if (theCase == WhichCase.lower)
                    System.out.println(line.toLowerCase());
                else if (theCase == WhichCase.upper)
                    System.out.println(line.toUpperCase());
                else
                    System.out.println(line);
            }

            fd.close();
        }
        catch (EOFException eof)
        {
            System.out.println("End of File (EOF) reached");
        }
        catch (IOException ioe)
        {
            System.out.println("IO error: " + ioe);
        }
    }
}

private enum WhichCase {unchanged, lower, upper};
}
```

Sample runs:

```
$ java type -u type.java
IMPORT JAVA.IO.BUFFEREDREADER;
IMPORT JAVA.IO.INPUTSTREAMREADER;

IMPORT JAVA.IO.FILEREADER;
```

```
IMPORT JAVA.IO.IOEXCEPTION;
IMPORT JAVA.IO.EOFEXCEPTION;
```

```
/* THIS PROGRAM TAKES ONE REQUIRED AND ONE OPTIONAL COMMAND LINE ARGUMENT:
```

```
THE FIRST ARGUMENT IS AN OPTIONAL SWITCH INDICATING WHETHER THE FILENAME IS TO BE
PRINTED OUT IN UPPER, OR LOWER CASE.
```

```
THE SECOND ARGUMENT IS IS REQUIRED; THIS ARGUMENT SPECIFIES THE NAME OF A FILE TO
BE PRINTED TO THE TERMINAL/TERMINAL WINDOW. THE CONTENTS OF THIS FILE ARE EITHER PRINTED OUT
UNCHANGED, UPPERCASE OR LOWER CASE DEPENDING UPON WHETHER THE FIRST, OPTIONAL ARGUMENT
IS SPECIFIED.
```

```
THE FOLLOWING IS HOW THIS PROGRAM IS EXPECTED TO BE USED:
```

```
    JAVA TYPE [-U|-L] FILENAME
```

```
NOTES:
```

- COMMAND LINE ARGUMENTS ARE SEPARATED BY SPACES
- COMMAND LINE ARGUMENTS STARTING WITH A DASH MUST BE LOWER CASE
- COMMAND LINE ARGUMENTS STARTING WITH A DASH, WHICH ARE SOMETIMES OPTIONAL, ARE CALLED "SWITCHES."
- THE EXPRESSION [-U|-L] MEANS THAT THE ARGUMENTS ARE OPTIONAL AND THAT EITHER A -U, A -L, OR A NO OPTION IS PERMITTED; HOWEVER, A -U AND A -L CANNOT BOTH BE USED TOGETHER

```
*/
```

```
PUBLIC CLASS TYPE
```

```
{
    PUBLIC STATIC VOID MAIN (STRING[] ARGS)
    {
        WHICHCASE THECASE = WHICHCASE.UNCHANGED; // UNCHANGED => OUTPUT FILE CONTENTS W/O CHANGING THE CASE
        FILEREADER FR = NULL;

        IF (ARGS.LENGTH < 1 || ARGS.LENGTH > 2)
        {
```

```
• • •
```

```
$ java type -p type.java
usage: java type [-u | -l] filename
```

```
$ java type type.java
import java.io.BufferedReader;
import java.io.InputStreamReader;

import java.io.FileReader;
```

```
import java.io.IOException;
import java.io.EOFException;
```

```
/* This program takes one required and one optional command line argument:
```

```
The first argument is an optional switch indicating whether the filename is to be
printed out in upper, or lower case.
```

```
The second argument is is required; this argument specifies the name of a file to
be printed to the terminal/terminal window. The contents of this file are either printed out
unchanged, uppercase or lower case depending upon whether the first, optional argument
is specified.
```

```
The following is how this program is expected to be used:
```

```
    java type [-u|-l] filename
```

```
Notes:
```

- Command line arguments are separated by spaces
- Command line arguments starting with a dash must be lower case
- Command line arguments starting with a dash, which are sometimes optional, are called "switches."
- The expression [-u|-l] means that the arguments are optional and that either a -u, a -l, or no option is permitted; however, a -u and a -l cannot both be used together

```
*/
```

```
public class type
```

```
{
    public static void main(String[] args)
    {
        WhichCase theCase = WhichCase.unchanged; // unchanged => output file contents w/o changing the case
        FileReader fr = null;

        if (args.length < 1 || args.length > 2)
        {
```

```
• • •
```

Note:

- The use of command line arguments
- Error processing
- The enum declaration and usage