

CSCI 1103: Command Line Arguments

Chris Kauffman

*Last Updated:
Thu Nov 30 11:25:17 CST 2017*

Command Line Argumnets

Pass command line arguments to a program

```
java CmdArgs here are 4 args
```

Use the arguments: stored in the args array of main

```
public class CmdArgs {
    public static void main(String args[]){
        if(args.length >= 4 &&
            args[0].equals("here") &&
            args[1].equals("are") &&
            args[2].equals("4") &&
            args[3].equals("args")){
            System.out.println("Hurray!");
        }
    }
}
```

See also `CommandLine.java`

Exercise: Check for repeated first

Write a `main()` method which checks if the 0th arg appears as any other argument.

```
> javac RepeatedFirst.java
```

```
> java RepeatedFirst
```

```
No words given
```

```
> java RepeatedFirst ding
```

```
ding not repeated
```

```
> java RepeatedFirst ding a ling
```

```
ding not repeated
```

```
> java RepeatedFirst ding a ling a ling ding a
```

```
ding repeated at position 5
```

```
> java RepeatedFirst ling ding a ling a ling
```

```
ling repeated at position 3
```

Answer: Check for repeated first

```
1 public class RepeatedFirst{
2     public static void main(String args[]){
3         if(args.length == 0){
4             System.out.println("No words given");
5             return;
6         }
7         String first = args[0];
8         for(int i=1; i<args.length; i++){
9             if(first.equals(args[i])){
10                System.out.printf("%s repeated at position %d\n",
11                                   first,i);
12                return;
13            }
14        }
15        System.out.printf("%s not repeated\n",first);
16    }
17 }
```

Command line args often name Files

- ▶ Outside world tells program which file(s) to work on
- ▶ Example: sum numbers in file named on command line

```
import java.io.*;
import java.util.*;

public class SumFile{
    public static void main(String args[]
    throws Exception
    {
        String filename = args[0];
        Scanner input =
            new Scanner(new File(filename));
        double sum = 0;
        while(input.hasNext()){
            double val = input.nextDouble();
            sum += val;
        }
        input.close();
        System.out.printf("sum of file %s: %.1f\n",
            filename,sum);
    }
}
```

```
> javac SumFile.java
> cat data.txt
17.0
18.0
16.0
10.0
15.0
> java SumFile data.txt
sum of file data.txt: 76.0
> cat data2.txt
7.0
8.0
6.0
0.0
5.0
4.5
6.5
> java SumFile data2.txt
sum of file data2.txt: 37.0
>
```

Exercise: Word Count

Write a `main()` method which

- ▶ Treats each command line arg as file name
- ▶ Opens each file using a Scanner
- ▶ Counts how many space-separated words appear in the file
- ▶ Closes the scanner
- ▶ Reports the results on screen
- ▶ Repeats for the next file in the args list

```
> javac WordCount.java
```

```
> java WordCount
```

```
> java WordCount WordCount.java
```

```
WordCount.java has 41 words
```

```
> java WordCount WordCount.java RepeatedFirst.java Example.java
```

```
WordCount.java has 41 words
```

```
RepeatedFirst.java has 39 words
```

```
Example.java has 137 words
```

Answer: Word Count

```
1 // Count all words in all files named on the command line
2
3 import java.util.Scanner;
4 import java.io.File;
5
6 public class WordCount{
7     public static void main(String args[]) throws Exception{
8         for(int i=0; i<args.length; i++){
9             String filename = args[i];
10            Scanner input = new Scanner(new File(filename));
11            int wc = 0;
12            while(input.hasNext()){
13                input.next();
14                wc++;
15            }
16            input.close();
17            System.out.printf("%s has %d words\n",
18                               filename,wc);
19        }
20    }
21 }
```