

CS120: The Command Line and Immutable Objects

W. Michael Petullo

University of Wisconsin–La Crosse

As of October 21, 2021





The Command Line: Shell

The shell is the program that provides you with a prompt on UNIX; the shell's most obvious use is to run other programs. We are using a variant of the Bourne shell.

```
cal
```

The shell can arrange to pass arguments to the programs it runs:

```
echo Hello , world!
```

The shell can also cause a program's output to redirect > to a file.

```
echo Hello , world! >hello
```

The shell has many more features; one of its most powerful features is to connect one program's output to another program's input. This is called a pipe (|):

```
echo Hello , world! | sed 's/Hello/Goodbye/g'
```

Shells allow for scripting.

The Command Line: Java Arguments



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

Java provides an array containing the program's command-line arguments to the `main` function. This is the parameter your `main` function has taken since you wrote "Hello, world!"

Know the difference between taking arguments from the command line and reading input from the keyboard!



Immutable Objects: Primitives vs. Objects

byte short int long float double boolean char

```
int number = -2;  
char symbol = '!';
```

number

-2

symbol

!

```
int [] array = { 1, 2, 3 };
```



Impacts

- ▶ comparing,
- ▶ copying, and
- ▶ passing to methods.

int []

double []

boolean []

char []

String

Scanner



Immutable Objects: null

You can assign `null`, which means “no object”, to an object variable.

```
int [] array = null;  
String s1 = null;  
String s2 = "";  
String s3 = "X";
```

array ○

s1 ○

s2 ○ → ""

s3 ○ → "X"

Java will raise a runtime error if you try to make use of an object that is set to `null`.

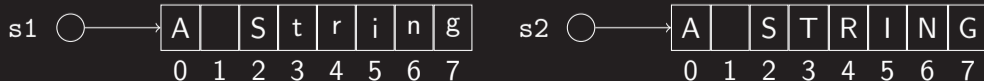
Immutable Objects: Unchangeable

Immutable objects cannot change after they come into existence. String s are immutable, so “modifications” to strings must create a new String.

```
String s1 = "A_String";  
String s2 = s1.toUpperCase();  
System.out.println(s1);  
System.out.println(s2);
```

prints

```
A String // Original string CANNOT change.  
A STRING
```





The Command Line: Assignments

Completed Chapter 7; completed Aquinas greets and swap in Java

Graded Homework Aquinas: factor in Java due next Monday

Ungraded Labs Aquinas: cli, area, and volume in Java

Reading Chapter 9

<https://www.flyn.org/courses/cs120-2021-fall/schedule>