Lesson00

Hello CS A!

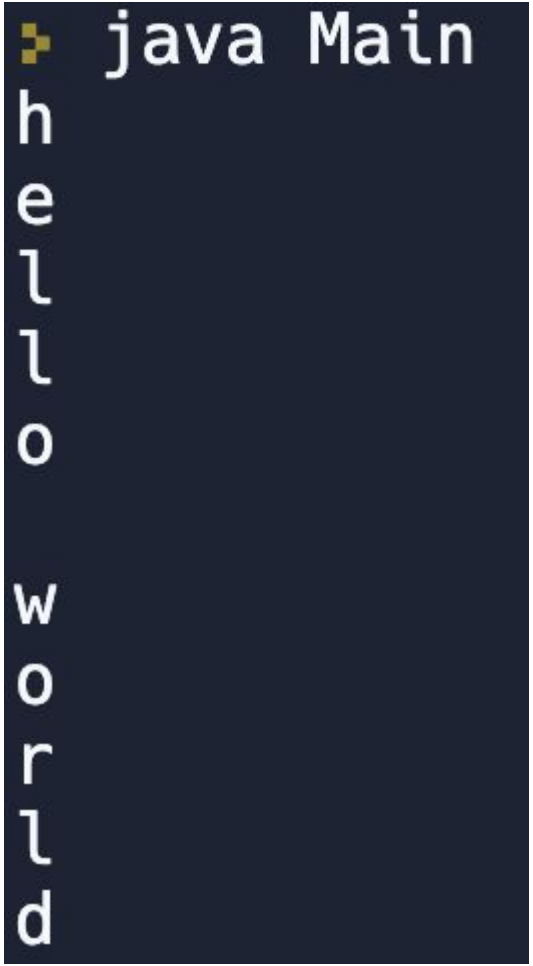
today’s starter activity:

```
String plaintext = “hello world”;
```

Have your program print out a String called plaintext, one character at a time.

```
// Hint
System.out.println("ABC".charAt(0)); // prints "A"

// Remember:
for (int i = 0; i < 10; i++){
  System.out.println(i);
}
// prints 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
```



```
> java Main
h
e
l
l
o
 
w
o
r
l
d
```

Caesar cipher

https://www.youtube.com/watch?v=l6jqKRXSShI

Code your own Caesar cipher!

DELIVERABLE:

A program that will shift each letter in a plaintext String by a certain amount of letters, as in a Caesar cipher.

It should be easy to change how many letters your program shifts by.

You can assume that the shifts will only be between -25 and 25.

Your program is only required to do one word each time that it is run.

Tests Your Code Should Pass:

- shifting "a" by 1 should be "b"
- shifting "b" by -1 should be "a"
- shifting "a" by -1 should be "z"
- shifting "z" by 1 should be "a"

Lesson 01

Hello CS A!

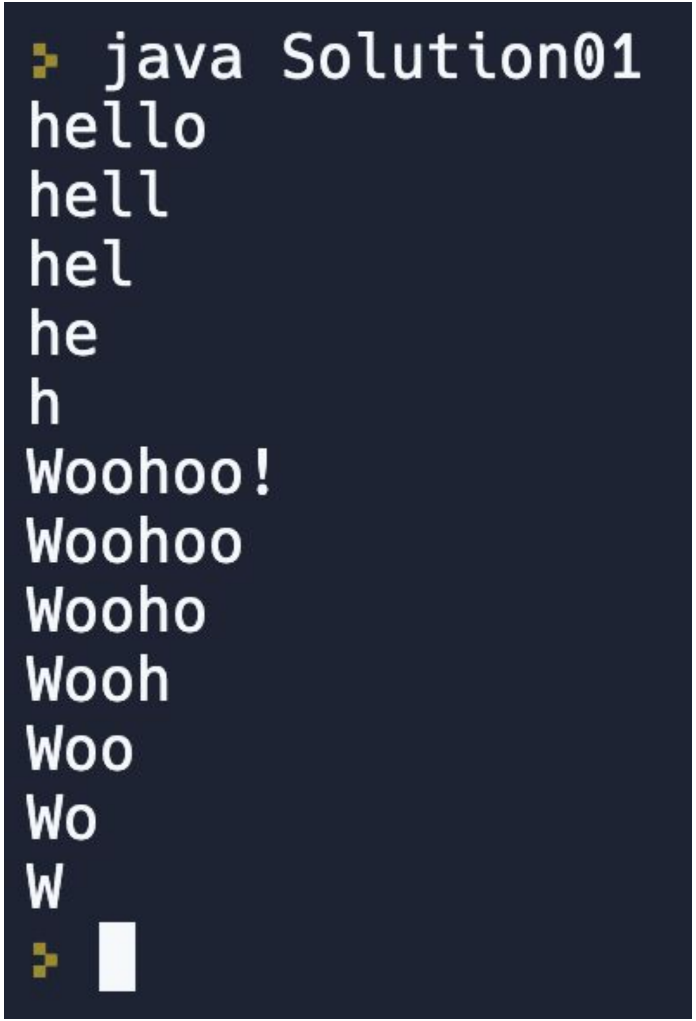
today’s starter activity:

```
public static void echoBack(String input)

echoBack(“hello”);
echoBack(“Woohoo!);
```

Write echoBack() however you would like to! Make sure that it produces the output on the right when it runs.

```
// Hint
String temp = "hello world!";
temp.substring(0, 2); // "he"
temp.substring(2, 5); // "llo"
```



```
> java Solution01
hello
hell
hel
he
h
Woohoo!
Woohoo
Wooho
Wooh
Woo
Wo
W
> █
```

Recursive function #1

```
public void foo(String input){
    System.out.println(input);
    foo(input);
}

public static void main(String[] args){
    foo("um");
}
```

Recursive function #2

```
public void foo(int input){
    System.out.println(input);
    foo(input - 1);
}

public static void main(String[] args){
    foo(10);
}
```

Recursive function #3

```
public void foo(int input){
    System.out.println(input);
    if (input < 100){
        foo(input + 1);
    }
}

public static void main(String[] args){
    foo(10);
}
```

Factorial, Fibonacci

```
factorial(3); // 3 * 2 * 1 = 6

factorial(4); // 4 * 3 * 2 * 1 = 24

factorial(1); // 1 = 1
```

Fibonacci sequence: 0 1 1 2 3 5 8 13 21 34 ...

1+1=**2**, 2+1=**3**, 2+3=**5**, 3+5=**8**, 5+8=**13**, 8+13=**21**, 13+21=**34**