IB Computer Science

Java Programming
Using Command Line Arguments

Student_____

Date: _____

Background

In all of the Java programs that we have used so far there has been a parameter to **`main`** that is an array of type String, such as:

```
public static void main (String[] args) {
```

When we have run Java programs we executed program java and passed the file specification that we wanted to run. The java program takes the command line arguments from the operating system and passes them to **`main`**. This is very similar to how we accepted command line arguments with C programs, the difference is that the C program also specified how many command line arguments there were (and we usually called this variable argsc with the string arguments being named argv).

Directions

1. Take a look at both programs Args01.java and Args02.java.
2. Read the Java Notes below
3. Follow the directions posted on today's Agenda on HWMath.net/IBCS.

Java Notes

```
int  argc = args.length;
for (int i = 0; i< argc; i++) {
```

In program Args01.java integer variable args is declared, and initialized to the length of the command line arguments array that we named args. The for loop will loop through each of the argc arguments starting with index 0. This should look familiar to you.

```
for (String s: args) {
```

In program Args02.java we loop through the args array a bit differently:
First notice that the loop variable s in being declared in the `for` statement.
The notice that the `for` statement has a little different format – it doesn't have three components, only one. This statement means "loop through each element in array args, copy the current argument that is being processed into variable s, and then perform all of the statements in the `for` block { }. This is equivalent to:

```
String s;
for (int i = 0; i < args.length; i++) {
     s = args[i];
     // do for-loop statements
```

In other programming languages you may have used, this would be similar to a "foreach" loop.

You might also notice that the output statements are slightly different. You can make changes as you see fit to have the programs produce identical ouput.

IB Computer Science

Java Programming
Using Command Line Arguments

Student_____

Date: _____

```
// Name: Mr. Brennan
// File: Args01.java
// Purpose: Echo each command line argument -
//          one argument per output line

public class Args01 {

  public static void main (String[] args) {

    String s="";
    int    argc = args.length;

    System.out.println("There are " + argc + " command line arguments:");

    for (int i = 0; i< argc; i++) {
        System.out.println("Argument #" + i + ": " + args[i]);
        } // end loop through the arguments

    } // end main

} // end Args01
```

```
// Name: Mr. Brennan
// File: Args02.java
// Purpose: Echo each command line argument -
//          one argument per output line

public class Args02 {

    public static void main (String[] args) {

        for (String s: args) {
            System.out.println(s);
        }

    } // end main

} // end Args02
```