Transitioning to Java
Transitioning from Processing to Java

How is it different from Processing?

• No built-in drawing, mouse/keyboard control
  – Functionality provided through additional libraries of code

• Execution begins at main()
  – Instead of setup()
  – No draw()

• Need to put everything inside a class
  – Also need additional modifiers (public/static)
Transitioning from Processing to Java

How is it different from Processing?

• Must use full names of library functions. For example:

  `println()` → `System.out.println()`
  `sin()` → `Math.sin()`
  `random()` → `Math.random()`
Transitioning from Processing to Java

What is the same? Just about everything else!

- variable types (int/float/double/String/etc.)
- conditionals (if, if/else)
- loops (for, while, break, continue)
- arrays
- functions
Hello World!
Your First Java Program

```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```
Your First Java Program

Program Name

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```
Your First Java Program

Scaffolding

```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```
Your First Java Program

Print the text "Hello, World".

```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```
Your First Java Program

Compile to translate to machine code

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```
Your First Java Program

Run the compiled program

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```
Command Line Arguments

• To run basic programs:
  > java HelloWorld

• Unlike Processing, Java programs can take in input on the command line via command line arguments
  > java HelloWorld Katie
Command Line Arguments

• What if we want the program say “Hello Katie” as opposed “Hello World”?
  – How would we change the current code?

```java
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello world!");
    }
}
```
Command Line Arguments

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

        String name = args[0];
        System.out.println("Hello there " +
                name);
    }
}
```
Command Line Arguments

• `args[0], args[1], ...` will be Strings

• How can we convert a String to an integer?
  – use the `Integer.parseInt()` function:

```java
int x;
String str = "37";
x = Integer.parseInt(str);
```
In-Class Exercise

• Write a program to compute exponentiation
  – Read in two integers a, b from the command line
  – Compute $a^b$ and print out the result

• Example usage:
  > run Power 3 2
    $3^2 = 9$
  > run Power 2 4
    $2^4 = 16$
Q. What happens when you compile and run the following code?

```java
public class Cubes1 {
    public static int cube(int i) {
        int j = i * i * i;
        return j;
    }

    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        for (int i = 1; i <= N; i++)
            System.out.println(i + " " + cube(i));
    }
}
```
Q. What happens when you compile and run the following code?

```java
public class Cubes1 {
    public static int cube(int i) {
        int j = i * i * i;
        return j;
    }

    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        for (int i = 1; i <= N; i++)
            System.out.println(i + " " + cube(i));
    }
}
```
Q. What happens when you compile and run the following code?

```java
public class Cubes1 {

    public static int cube(int i) {
        int j = i * i * i;
        return j;
    }

    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        for (int i = 1; i <= N; i++)
            System.out.println(i + " " + cube(i));
    }
}
```

% run Cubes1
1 1
2 8
3 27
4 64
5 125
6 216
Q. What happens when you compile and run the following code?

```java
public class Cubes2 {
    public static int cube(int i) {
        int i = i * i * i;
        return i;
    }

    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        for (int i = 1; i <= N; i++)
            System.out.println(i + " " + cube(i));
    }
}
```
Q. What happens when you compile and run the following code?

```java
public class Cubes3 {
    public static int cube(int i) {
        i = i * i * i;
    }

    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        for (int i = 1; i <= N; i++)
            System.out.println(i + " " + cube(i));
    }
}
```
Q. What happens when you compile and run the following code?

```java
public class Cubes4 {

    public static int cube(int i) {
        i = i * i * i;
        return i;
    }

    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        for (int i = 1; i <= N; i++)
            System.out.println(i + " " + cube(i));
    }
}
```
Q. What happens when you compile and run the following code?

```java
public class Cubes5 {
    public static int cube(int i) {
        return i * i * i;
    }
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        for (int i = 1; i <= N; i++)
            System.out.println(i + " " + cube(i));
    }
}
```
Top Down Design Practice

• write a program that, given an array, creates a copy:
  1. that is one element larger than the original
  2. that puts the sum in the last element

<table>
<thead>
<tr>
<th>original</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
</table>

| copy     | 1 | 1 | 2 | 3 | 7 |

• Hint: First think about what functions you will need.