

1 Our First Java Program

Below is our first Java program of the semester. Next to each line, write out what you think the code will do when run.

Comments are above the line they describe

```
1      /* Declare a variable of type String and assign it the value "no". In  
        Java, all variables must be declared before they are used. */  
2      String disagree = "no";  
3  
4      /* Declare a variable of type int and assign it the value 7. */  
5      int x = 7;  
6  
7      /* Declare a variable of type Singer and initialize it using the Singer  
        constructor with the argument "Queen" */  
8      Singer queen = new Singer("Queen");  
9  
10     /* Checks if x is greater than 0; if so, subtract 1 from x, then call  
        queen's sing method with argument "no", then go back to the beginning  
        of the loop. queen object sings "no" 7 times. */  
11     while (x > 0) {  
12         x -= 1;  
13         queen.sing(disagree);  
14     }  
15  
16     /* Declares a variable of type String array and initializes it to hold  
        three Strings. */  
17     String[] phrases = {"Oh", "mamma mia", "let me go"};  
18  
19     /* Prints "Oh" to standard output */  
20     System.out.print(phrases[0]);  
21  
22     /* Declares variable i and initialize to 0 and checks if it is less than  
        3. If so, print "mamma mia", then add one to i, then go back to the  
        beginning of the loop and re-check that i < 3. "mamma mia" is printed  
        out 3 times. */  
23     for (int i = 0; i < 3; i += 1) {  
24         System.out.print(" " + phrases[1]);  
25     }  
26  
27     /* Prints "let me go" to standard output. */  
28     System.out.print(" " + phrases[2]);
```

2 Writing Your First Program

```
/** fib(N) returns the Nth Fibonacci number, for N ≥ 0.  
    * The Fibonacci sequence is 0, 1, 1, 2, 3, 5, 8, 13, 21, ... */
```

```

public static int fib(int N) {
    if (N <= 1) {
        return N;
    } else {
        return fib(N - 1) + fib(N - 2);
    }
}

```

We can also write this iteratively:

```

/** fib(N) returns the Nth Fibonacci number, for N ≥ 0.
 * The Fibonacci sequence is 0, 1, 1, 2, 3, 5, 8, 13, 21, ... */
public static int fib(int N) {
    int f0 = 0;
    int f1 = 1;
    while (N > 0) {
        int temp = f1;
        f1 = f0 + f1;
        f0 = temp;
        N -= 1;
    }
    return f0
}

```

Extra for experts: Complete fib2 in 5 lines or less. Your answer must be efficient.

```

public static int fib2(int n, int k, int f0, int f1) {
    if (n == k) {
        return f0;
    } else {
        return fib2(n, k + 1, f1, f0 + f1);
    }
}

```

To compute the N^{th} fibonacci number using fib2, call fib2(N, 0, 0, 1).

3 Mystery

```

1  /** This is a function (a.k.a. method). It takes an array
2     * of integers as an argument, and returns an integer. */
3  public static int mystery(int[] inputArray, int k) {
4     int x = inputArray[k];
5     int answer = k;
6     int index = k + 1;
7     while (index < inputArray.length) {
8         if (inputArray[index] < x) {
9             x = inputArray[index];
10            answer = index;
11        }
12        index = index + 1;
13    }
14    return answer;
15 }
16
17  /** Extra for experts. This is another function. It takes an

```

```

18     * array of integers and returns nothing at all. */
19     public static void mystery2(int[] inputArray) {
20         int index = 0;
21         while (index < inputArray.length) {
22             int targetIndex = mystery(inputArray, index);
23             int temp = inputArray[targetIndex];
24             inputArray[targetIndex] = inputArray[index];
25             inputArray[index] = temp;
26             index = index + 1;
27         }
28     }

```

- What does `mystery` return if `inputArray` is the array 3, 0, 4, 6, 3, and `k` is 2?

It returns 4.

- Describe, in English, what `mystery` returns.

It returns the index of the smallest element that occurs at or after index `k` in the array. If `k` is greater than or equal to the length of the array or less than 0, an `ArrayIndexOutOfBoundsException` will be thrown, though this exception is not something you'd know without running the program.

The variable `x` keeps track of the smallest element found so far and the variable `answer` keeps track of the index of this element. The variable `index` keeps track of the current position in the array. The while loop steps through the elements of the array starting from index `k + 1` and if the current element is less than `x`, `x` and `answer` are updated.

- Extra for experts: What does `mystery2` do if `inputArray` is the array 3, 0, 4, 6, 3? Describe, in English, what `mystery2` does to the array.

If `mystery2` is called on the array 3, 0, 4, 6, 3 then after the method runs, the array will be 0, 3, 3, 4, 6. Given any array, the method `mystery2` sorts the elements of the array in increasing order.

At the beginning of each iteration of the while loop, the first `index` elements of the array are in sorted order. Then the method `mystery` is called to find the index of the smallest element of the array occurring at or after `index`. The element at the index returned by `mystery` is then swapped with the element at position `index` so that the first `index + 1` elements of the array are in sorted order.