

## 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. *This exercise is adapted from Head First Java.*

```
1  int size = 27;
2  // Declares a variable of type int and assigns it the value 27. In Java, all variables must be
3  // declared before they are used
4  String name = "Fido";
5  // Declares a variable of type String and assigns it the variable "Fido"
6  Dog myDog = new Dog(name, size);
7  // Declares and initializes a new variable of type Dog. Calls the Dog constructor to create a new
8  // object of type Dog
9  int x = size - 5;
10 // Declares a new variable of type int and assigns it the value 22
11 if (x < 15) {
12 // If x is less than 15, calls the bark method on an instance of the Dog class. Since x is 22,
13 // myDog.bark is not called
14     myDog.bark(8);
15 }
16
17 while (x > 3) {
18 // Checks if x is greater than 3 and if so calls myDog's play method. Subtracts 1, and as long
19 // as x is bigger than 3, goes back to the beginning of the loop. Play happens a total of 19 times.
20     x -= 1;
21     myDog.play();
22 }
23
24 int[] numList = {2, 4, 6, 8};
25 // Declares an array of ints and initializes it to {2, 4, 6, 8}
26 System.out.print("Hello ");
27 // Prints the String "Hello " to the standard output
28 System.out.println("Dog: " + name);
29 // Prints the String "Dog: Fido" to the standard output and then terminates the line
30 System.out.println(numList[1]);
31 // Prints the String "4" to the standard output and then terminates the line. In Java, arrays are
32 // indexed from 0
33 if (numList[3] == 8) {
34 // numList[3] is equal to 8
35     System.out.println("potato");
36 // Prints the String "potato" to the standard output and then terminates the line
37 }
```

For your convenience, here is the same code in Python:

```

1 size = 27
2 name = "Fido"
3 myDog = Dog(name, size)
4 x = size - 5;
5 if x < 15:
6     myDog.bark(8)
7 while x > 3:
8     x -= 1
9     myDog.play()
10 numList = [2, 4, 6, 8]
11 print("Hello")
12 print("Dog: " + name)
13 print(numList[1])
14 if numList[3] == 8:
15     print("potato")

```

## 2 Mystery

This is a function (a.k.a. method). It takes an array of integers and an integer as arguments, and returns an integer.

```

1 public static int mystery(int[] inputArray, int k) {
2     int x = inputArray[k];
3     int answer = k;
4     int index = k + 1;
5     while (index < inputArray.length) {
6         if (inputArray[index] < x) {
7             x = inputArray[index];
8             answer = index;
9         }
10        index = index + 1;
11    }
12    return answer;
13 }

```

(a) What `mystery` returns if `inputArray = [3, 0, 4, 6, 3]` and `k = 2`.

The method returns 4.

(b) Can you explain in English what does `mystery` do?

It returns the index of the smallest element that occurs at or after index `k` in the array, in this case, 4. 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:* This is another function. It takes an array of integers and returns nothing.

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputArray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }
```

Describe what `mystery2` does if `inputArray = [3, 0, 4, 6, 3]`.

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. This algorithm is called **selection sort**. We will talk about it more later on in the course.

### 3 Writing Your First Program

Implement `fib` which takes in an integer  $n$  and returns the  $n$ th Fibonacci number.

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);
    }
}
```

*Extra:* Implement `fib` in 5 lines or fewer. Your answer must be efficient. You don't have to make use of the parameter `k` in your solution.

```

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