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.

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

Acknowledgement: This exercise is adapted from page 5 of our textbook Head First Java.
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.

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

3 Writing Your First Program

```java
/** fib(n) returns the n\text{th} Fibonacci number, for n\geq0. 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);
    }
}
```

This solution above is recursive. We can also write it iteratively:

```java
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.

```java
/** fib2(n, 0, 0, 1) returns the n\text{th} Fibonacci number, for n\geq0. */
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);
    }
}
```

This is the tail-recursive solution. We can write this more succinctly using the ternary operator:

```java
/** fib2(n, 0, 0, 1) returns the n\text{th} Fibonacci number, for n\geq0. */
public static int fib2(int n, int k, int f0, int f1) {
    return n == k ? f0 : fib2(n, k + 1, f1, f0 + f1);
}
```