

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;      //declare an int variable called size and initiate it with 27
2 String name = "Fido"; //declare a String variable called name that has reference to a String object that stores "Fido"
3 Dog myDog = new Dog(name, size); //create a Dog object called myDog and initialize its fields with name and size using constructor
4 int x = size - 5; //declare an int variable called x and initialize with a value of size-5
5 if (x < 15) {
6     myDog.bark(8); //Test the condition x<15,
7 }                   // if the condition is true, call the bark() method of myDog object and pass 8 as method's parameter
8                     // In this case x=22 > 15, the method will not be called
9 while (x > 3) {   // Keep executing x = x-1 and calling the play() method of myDog object until x <=3
10    x -= 1;        // In this case, the loop will execute 19 times.
11    myDog.play(); // In this case, the loop will execute 19 times.
12 }
13
14 int[] numList = {2, 4, 6, 8}; //create an int array called numList in which stores 2, 4, 6, 8
15 System.out.print("Hello "); //print out "Hello"
16 System.out.println("Dog: " + name); //print out "Dog: Fido" and then go to a new line
17
18 System.out.println(numList[1]); //print out the second item in the numList array and then go to a new line
19 if (numList[3] == 8) {
20     System.out.println("potato"); //Test whether the fourth item in the numList is equal to 8
21 }                           //If it is, print out "potato", in this case x is equal to 8, and it prints out "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 }
```

Describe in English what `mystery` returns if `inputArray = [3, 0, 4, 6, 3]` and `k = 2`.

The `mystery` function is used to find and print out the index of the minimum value of the `inputArray` starting from the `k`th position. In this case, 4 is returned.

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]`.

The `mystery2` function sorts the `inputArray` from smallest to largest. When we call the function with the given `inputArray`, the `inputArray` will be sorted to be `[0,3,4,6]`.

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==0){  
        return 0;  
    }else if(n==1){  
        return 1;  
    }else{  
        return fib(n-1)+fib(n-2);  
    }  
}
```

}

Extra: Implement `fib` in 5 lines or fewer. Your answer must be efficient.

```
public static int fib2(int n, int k, int f0, int f1) {  
    /*I think that n indicates the nth Fibonacci number we want to get. k indicates the starting position of counting.  
    f0 indicates the Fibonacci number of the starting point. f1 indicates the Fibonacci number of index k+1.  
    For example, fib2(3,1,1,1) indicates that starts from index 1. This function enables us to choose the starting point.  
    This reduces the times the function is recursively called.*/  
    if(n==k){  
        return f0;  
    }else{  
        return fib2(n,k+1,f1,f0+f1);  
    }  
}
```