

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. If you think the line will result in an error, correct it, and proceed through the code as if it is running your corrected version.

This exercise is adapted from Head First Java.

```
1  size = 27;           Error, type not declared. Correction: int size = 27
2  name = "Fido";       Error, type not declared. Correction: String name = "Fido"
3  Dog myDog = new Dog(name, size);  Declares and initialises an instance of Dog "myDog"
4  Dog yourDog = new Dog("Scruffy", 1000);  Declares and initialises an instance of Dog "yourDog"
5  Dog[] dogList = new Dog[3];  Declares a list "dogList" that will contain 4 Dog objects
6  dogList[0] = myDog;           myDog added to dogList
7  dogList[1] = yourDog;         yourDog added to dogList
8  dogList[2] = 5;               TypeError, cannot add integers to dogList as dogList can only hold Dog objects
9  dogList[3] = new Dog("Cutie", 8)  valid, Dog new Dog added to the list
10 int x;                       valid, initialised int x with no value
11 x = size - 5;                 x = 22
12 if (x < 15) {                 False, don't execute code inside
13     myDog.bark(8);             skipped, but there will be a compilation error if the Dog
14 }                             class does not have a bark method or if the method
                                signature accepts a different argument to a single int
```

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

Text

It would return 4, this is the final value of index. You can see my workings where red is the first loop iteration, blue is the second iteration

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

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

The function returns the position of the smallest value in the array after position k. If no such value, it returns position k

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].

mystery2 sorts inputArray smallest to largest by looking for smallest value after current index. If it finds a smaller value, it assigns this targetIndex, it swaps the position of currentIndex with targetIndex

According to the web, this is called a selection sort

3 Writing Your First Program

Implement `fib` which takes in an integer `n` and returns the n th Fibonacci number. You may not need to use all the lines.

The Fibonacci sequence is 0, 1, 1, 2, 3, 5, 8, 13, 21, ... The first two numbers in the sequence are 0 and 1, and every number thereafter it is the sum of the two numbers in the sequence before it.

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

Extra: Implement a more efficient version of `fib` in 5 lines or fewer. Here, efficiency might mean making less recursive calls or doing less overall computation. You don't have to make use of the parameter `k` in your solution.

```
public static int fib2(int n, int k, int f0, int f1) {
    if (n == 0) return f0;
    else if (n == 1) return f1;
    else return fib2(n-1, f1, f0+f1);
}
```