

Name:

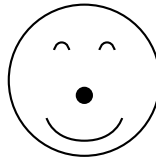
SID:

CS 61B Discussion Quiz 1

Write your name and SID above. Detach this page from your discussion handout, and turn it in when your TA instructs you to do so. **These quizzes are used as attendance.**

Questions

1. Tell us a little about yourself.
My name is CS 61B. I like data structures.
2. What do you expect to learn from the course?
Who *is* ketchupfriend?
3. Please draw how you feel today in the box below. :)



This page is intentionally left blank.

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

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`.

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) {
```

[0.75in]

```

1     if (n <= 1) {
2         return n;
3     } else {
4         return fib(n - 1) + fib(n - 2); //嵌套
5     }
6 }

```

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) {  
1      if (n == k) {  
2          return f0;  
3      } else {  
4          return fib2(n, k + 1, f1, f0 + f1);  
5      }  
6  }
```

这段代码是一个递归函数实现，用于计算第 n 个斐波那契数。

变量解释：

n : 这是函数 fib2 的参数，表示要计算的斐波那契数的索引，即第 n 个斐波那契数。

k : 这是递归调用中的辅助参数，用于迭代计数。初始时， k 通常为 0 或 1，取决于斐波那契序列的起始索引。

$f0$: 这是斐波那契数列中当前计算的前一个数的值。

$f1$: 这是斐波那契数列中当前计算的最后一个数的值。

函数说明：

fib2(int n , int k , int $f0$, int $f1$): 这是一个递归函数，用于计算第 n 个斐波那契数。

基础情况 ($n == k$): 当 n 等于 k 时，表示已经递归到达目标斐波那契数的索引位置。此时函数返回 $f0$ ，即当前斐波那契数列中的值。

递归情况：当 n 不等于 k 时，递归调用 fib2 函数，参数为 (n , $k + 1$, $f1$, $f0 + f1$)。这里的参数更新如下：

n 不变，表示计算的目标斐波那契数仍然是第 n 个。

$k + 1$ ，表示递归迭代到下一个位置。

$f1$ ，作为下一次递归调用中的 $f0$ ，即当前斐波那契数列的前一个数。

$f0 + f1$ ，作为下一次递归调用中的 $f1$ ，即当前斐波那契数列的最后一个数，是前两个数的和。