welcome!

to cs61b

cameras on if you can! 

slides
bit.ly/abhi-disc

attendance
bit.ly/abhi-attendance

about me

about me

- abhi (he/him/his)

about me

- abhi (he/him/his)
- from st. louis, missouri

about me

- abhi (he/him/his)
- from st. louis, missouri
- youtube and netflix consume a lot of my life

about me

- abhi (he/him/his)
- from st. louis, missouri
- youtube and netflix consume a lot of my life
- i’m here to be your point of contact!
  - Disc: M 4-5pm Soda 310
  - Lab: F 11am-1pm Soda 271
  - abhiganesh@berkeley.edu

about you

- name, pronouns, major, year, anything
- where are you from?
- thoughts on cs61a/coding/CS (👍, 👎, 😵‍💫)
- misc
  - favorite food
  - best places to visit in berkeley
  - hobbies
  - favorite power ranger

announcements

announcements

1. Lab 1, Lab 2, and HW 0 due Friday 01/28 (all of these CANNOT be dropped)

announcements

1. Lab 1, Lab 2, and HW 0 due Friday 01/28 (all of these CANNOT be dropped)
2. HW 1 released Tuesday at noon, due next Tuesday 02/01

announcements

1. Lab 1, Lab 2, and HW 0 due Friday 01/28 (all of these CANNOT be dropped)
2. HW 1 released Tuesday at noon, due next Tuesday 02/01
3. OH starts this week entirely online

announcements

1. Lab 1, Lab 2, and HW 0 due Friday 01/28 (all of these CANNOT be dropped)
2. HW 1 released Tuesday at noon, due next Tuesday 02/01
3. OH starts this week entirely online
4. Please complete the Pre-Semester Survey!

```java
/** Traditional first program.
 * @author P. N. Hilfinger */
public class Hello {
   /** Print greeting. ARGS is ignored. */
   public static void main(String[] args) {
      System.out.println("Hello, world!");
   }
}
```

```
/** Traditional first program.
 * @author P. N. Hilfinger */
public class Hello {
   /** Print greeting. ARGS is ignored. */
   public static void main(String[] args) {
      System.out.println("Hello, world!");
   }
}
```

documentation comment

```
/** Traditional first program.
 * @author P. N. Hilfinger */
public class Hello {
   /** Print greeting. ARGS is ignored. */
   public static void main(String[] args) {
      System.out.println("Hello, world!");
   }
}
```

documentation comment

comment

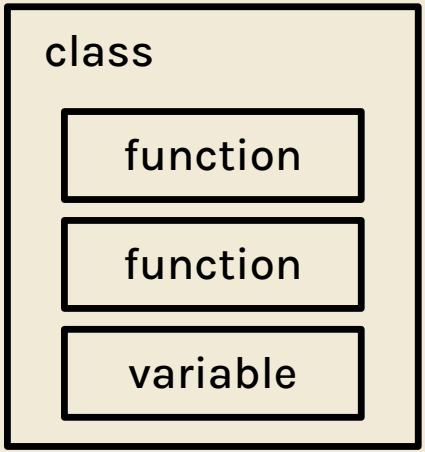
```
/** Traditional first program.
 * @author P. N. Hilfinger */
public class Hello {
   /** Print greeting. ARGS is ignored. */
   public static void main(String[] args) {
      System.out.println("Hello, world!");
   }
}
```

class

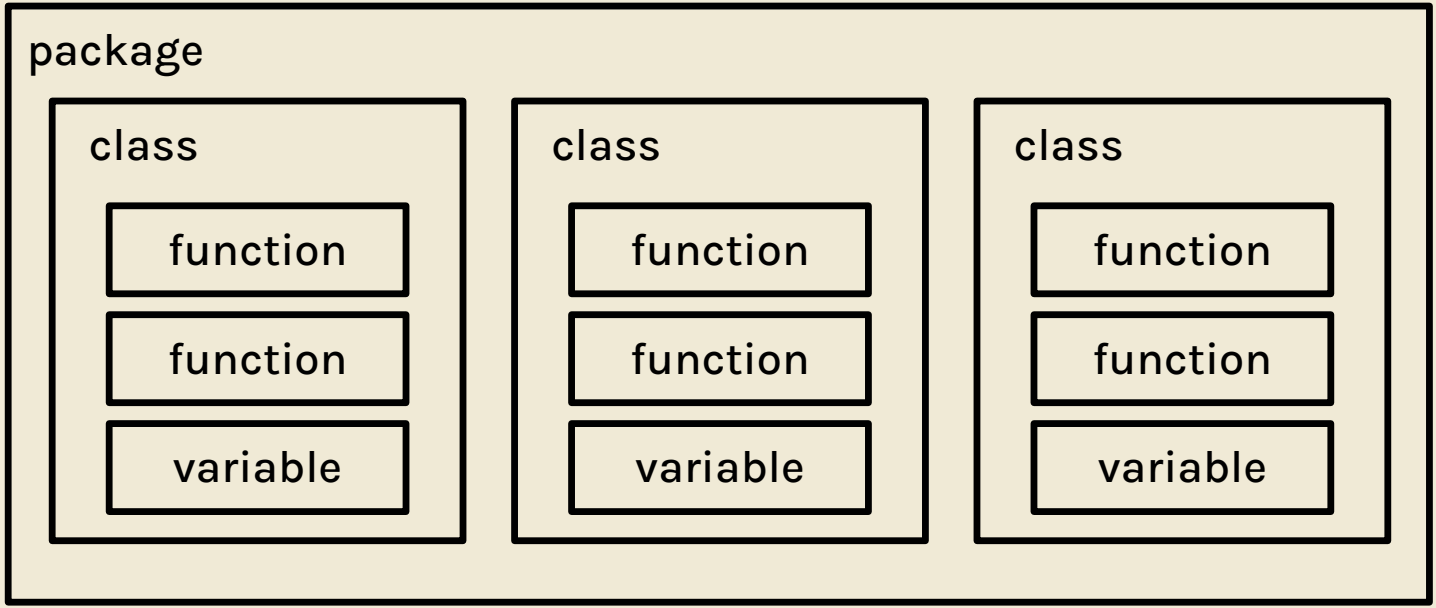
classes...?

classes

- contain all functions/variables in java
- classes belong to a package



classes


package
class
function
function
variable
class
function
function
variable
class
function
function
variable

structure

```
public class CS61BStudent { // Class Declaration
    public int idNumber; // Instance Variables
    public int grade;

    public static String professor = "Hilfinger"; // Class (Static) Variables

    public CS61BStudent (int id) { // Constructor
        this.idNumber = id;
        this.grade = 100;
    }

    public void watchLecture() { // Instance Method
        ...
    }

    public static void updateGrades() { // Class (Static) Method
        ...
    }
}
```

instantiation

```
public class CS61BLauncher {
    public static void main(String[] args) {
        CS61BStudent studentOne; // Declare class
        studentOne = new CS61BStudent(32259); // Instantiate and assign class
        CS61BStudent studentTwo = new CS61BStudent(19234); // Both at once

        studentOne.watchLecture(); // Instance methods are called on instance

        CS61BStudent.updateGrades(); // Static methods can be
                                     // called by class OR instance
    }
}
```

static

- belongs to whole class
  - All 61B students share the same Prof.

instance

- belongs to individual instance
  - Each student has their own SID

```java
/** Traditional first program.
 * @author P. N. Hilfinger */
public class Hello {
   /** Print greeting. ARGS is ignored. */
   public static void main(String[] args) {
      System.out.println("Hello, world!");
   }
}
```

the function header

function name

```
public static void main(String[] args) {
    System.out.println("Hello, world!");
}
```

the function header

function name

parameter(s)

```
public static void main(String[] args) {
    System.out.println("Hello, world!");
}
```

the function header

type

```
public static void main(String[] args) {
    System.out.println("Hello, world!");
}
```

the function header

public class CS61BStudent {

3

CS61BS

public ~~int~~ greeted() {

return type

parameter type

3

```
public static void main(String[] args) {

    System.out.println("Hello, world!");

}
```

int 5

String "hi"

double (26)

boolean T/F

char 'a'

worksheet

(on 61B website)

1 Old Town Code

```
int x = 7;
String chorus = “Thank u, next”;
Singer queen = new Singer(“Ariana”);

while (x > 0) {
    x -= 1;
    queen.sing(chorus);
}

String[] phrases = {“love”, “patience”, “pain”, “what does the fox say?”};

for (int i = 0; i < 3; i += 1) {
    System.out.println(“One taught me “ + phrases[i]);
}

System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
int x = 7; // Declares var x and assigns 7 to it
String chorus = “Thank u, next”;
Singer queen = new Singer(“Ariana”);

while (x > 0) {
    x -= 1;
    queen.sing(chorus);
}

String[] phrases = {“love”, “patience”, “pain”, “what does the fox say?”};

for (int i = 0; i < 3; i += 1) {
    System.out.println(“One taught me “ + phrases[i]);
}

System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
int x = 7; // Declares var x and assigns 7 to it
String chorus = “Thank u, next”; // Declares var chorus and assigns a String to it
Singer queen = new Singer(“Ariana”);

while (x > 0) {
    x -= 1;
    queen.sing(chorus);
}

String[] phrases = {“love”, “patience”, “pain”, “what does the fox say?”};

for (int i = 0; i < 3; i += 1) {
    System.out.println(“One taught me “ + phrases[i]);
}

System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
int x = 7; // Declares var x and assigns 7 to it
String chorus = “Thank u, next”; // Declares var chorus and assigns a String to it
Singer queen = new Singer(“Ariana”); // Declares var queen and assigns a Singer to it

while (x > 0) {
    x -= 1;
    queen.sing(chorus);
}

String[] phrases = {“love”, “patience”, “pain”, “what does the fox say?”};

for (int i = 0; i < 3; i += 1) {
    System.out.println(“One taught me “ + phrases[i]);
}

System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
int x = 7; // Declares var x and assigns 7 to it
String chorus = “Thank u, next”; // Declares var chorus and assigns a String to it
Singer queen = new Singer(“Ariana”); // Declares var queen and assigns a Singer to it

while (x > 0) { // Checks if x is still greater than 0
    x -= 1; // If so it deducts 1 from x
    queen.sing(chorus); // And it calls the queen.sing method on chorus
}

String[] phrases = {“love”, “patience”, “pain”, “what does the fox say?”};

for (int i = 0; i < 3; i += 1) {
    System.out.println(“One taught me “ + phrases[i]);
}

System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
int x = 7; // Declares var x and assigns 7 to it
String chorus = “Thank u, next”; // Declares var chorus and assigns a String to it
Singer queen = new Singer(“Ariana”); // Declares var queen and assigns a Singer to it

while (x > 0) { // Checks if x is still greater than 0
    x -= 1; // If so it deducts 1 from x
    queen.sing(chorus); // And it calls the queen.sing method on chorus
}

String[] phrases = {“love”, “patience”, “pain”, “what does the fox say?”};
// Declares var phrases and assigns an array of Strings to it
for (int i = 0; i < 3; i += 1) {
    System.out.println(“One taught me “ + phrases[i]);
}

System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
int x = 7; // Declares var x and assigns 7 to it
String chorus = “Thank u, next”; // Declares var chorus and assigns a String to it
Singer queen = new Singer(“Ariana”); // Declares var queen and assigns a Singer to it

while (x > 0) { // Checks if x is still greater than 0
    x -= 1; // If so it deducts 1 from x
    queen.sing(chorus); // And it calls the queen.sing method on chorus
}

String[] phrases = {“love”, “patience”, “pain”, “what does the fox say?”};
// Declares var phrases and assigns an array of Strings to it
for (int i = 0; i < 3; i += 1) { // Declares i and checks if its still less than 3
    System.out.println(“One taught me “ + phrases[i]); // If so it prints
} // And increments by one

System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
int x = 7; // Declares var x and assigns 7 to it
String chorus = “Thank u, next”; // Declares var chorus and assigns a String to it
Singer queen = new Singer(“Ariana”); // Declares var queen and assigns a Singer to it

while (x > 0) { // Checks if x is still greater than 0
    x -= 1; // If so it deducts 1 from x
    queen.sing(chorus); // And it calls the queen.sing method on chorus
}

String[] phrases = {“love”, “patience”, “pain”, “what does the fox say?”};
// Declares var phrases and assigns an array of Strings to it
for (int i = 0; i < 3; i += 1) { // Declares i and checks if its still less than 3
    System.out.println(“One taught me “ + phrases[i]); // If so it prints
} // And increments by one

System.out.println(phrases[phrases.length - 1]); // Prints the last phrase in phrases
```

What does this output?

1 Old Town Code

```
int x = 7;
String chorus = “Thank u, next”;
Singer queen = new Singer(“Ariana”);

while (x > 0) {
    x -= 1;
    queen.sing(chorus);
}

String[] phrases = {“love”, “patience”, “pain”, “what does the fox say?”};

for (int i = 0; i < 3; i += 1) {
    System.out.println(“One taught me “ + phrases[i]);
}

System.out.println(phrases[phrases.length - 1]);
```

Console Output

```
One taught me love
One taught me patience
One taught me pain
What does the fox say?
```

What does this output?

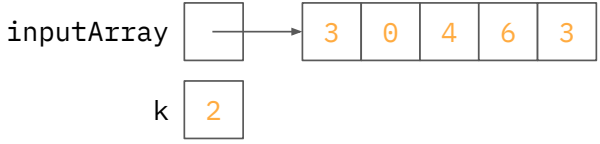
2 Reading Code: A Mystery

```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```

What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

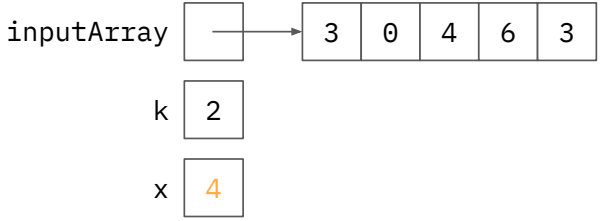
```java
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

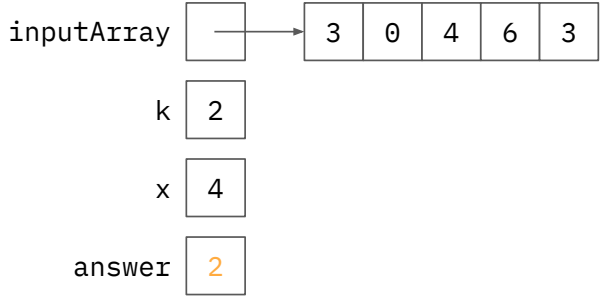
```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

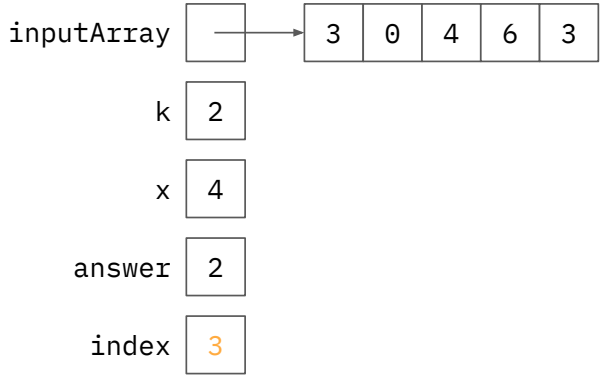
```java
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

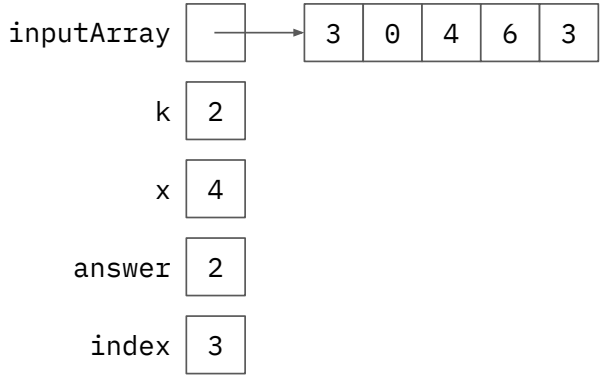
```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

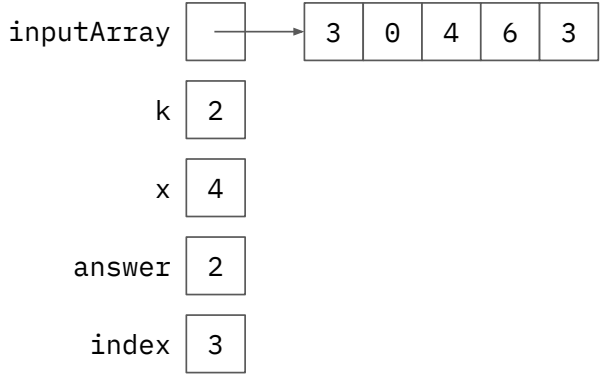
```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) { // True
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

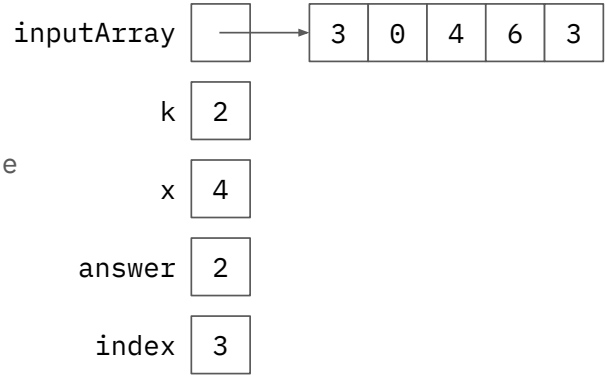
```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) { // False
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

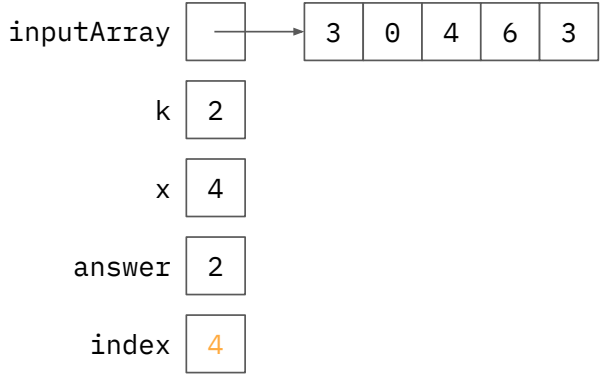
```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) {
            x = inputArray[index]; // Skip this line
            answer = index; // and this one
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

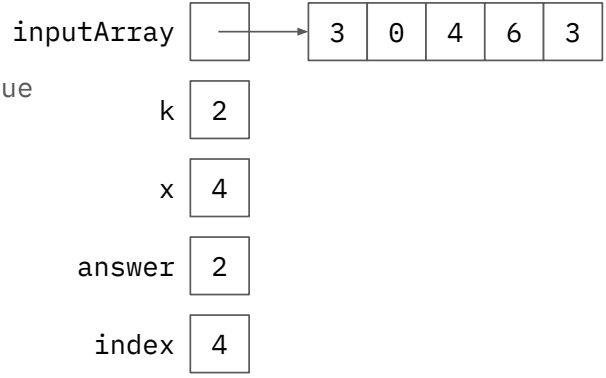
```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

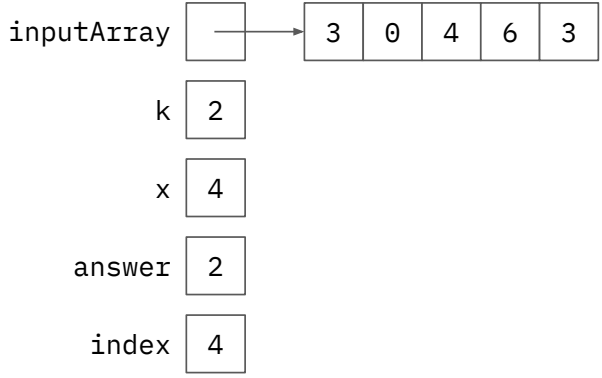
```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) { // still True
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

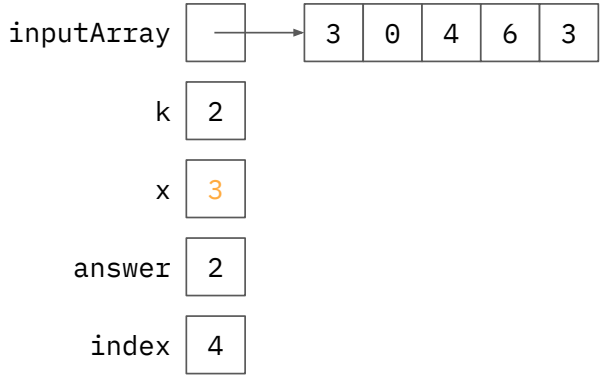
```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) { // True
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

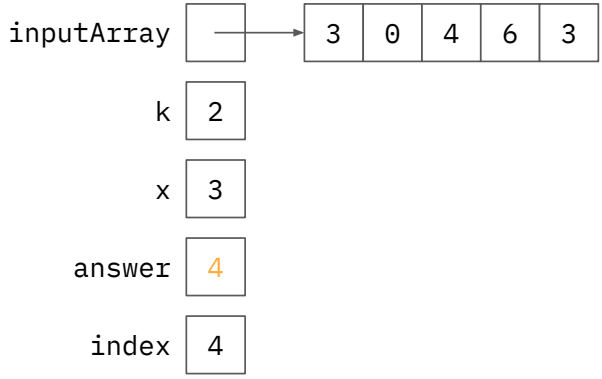
```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

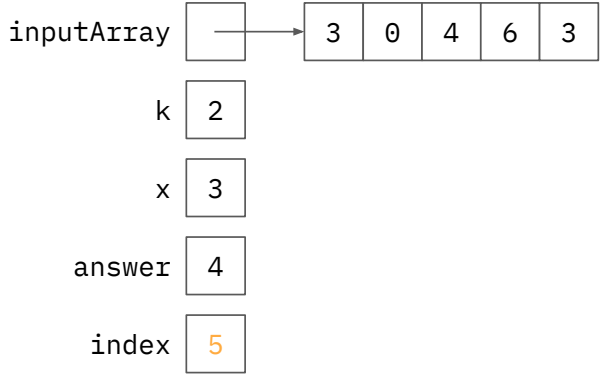
```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

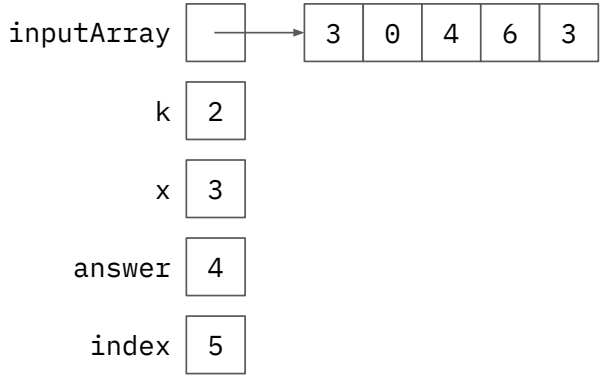
```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

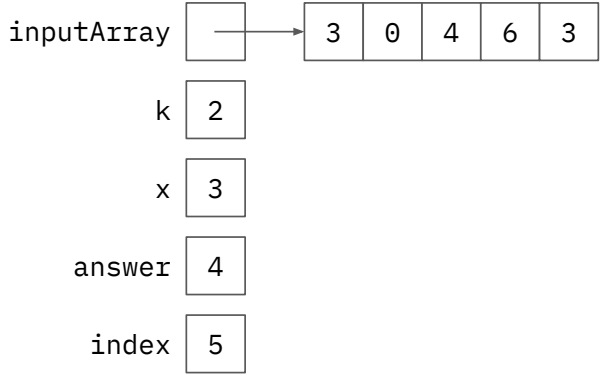
```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) { // False
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) { // Skip
            x = inputArray[index]; // all
            answer = index; // of
        } // these
        index = index + 1; // lines
    }
    return answer;
}
```



What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

2 Reading Code: A Mystery

```
public static int mystery1(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```

Return: 4

The returned value is the index of the smallest value in the array that occurs at or after index k.

What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?

3 Recursion Practice: Fibonacci

Implement a function `fib1` that recursively calculates the `N`th fibonacci number.

Hint: `fib(N) = fib(N-1) + fib(N-2)`

1, 1, 2, 3, 5

```
public static int fib1(int N) {
        if(N==0 || N==1) {
            return 1
        } else {
            return fib1(N-1) + fib1(N-2);
        }
}
```

3 Recursion Practice: Fibonacci

Implement a function `fib1` that recursively calculates the `N`th fibonacci number.

Hint: `fib(N) = fib(N-1) + fib(N-2)`

```
public static int fib1(int N) {
    if (N <= 1) { // Base case - can be written a few different ways
        return N;
    }


}
```

3 Recursion Practice: Fibonacci

Implement a function `fib1` that recursively calculates the `N`th fibonacci number.

Hint: `fib(N) = fib(N-1) + fib(N-2)`

```
public static int fib1(int N) {
    if (N <= 1) {
        return N;
    } else {
        return fib1(N - 1) + fib2(N - 2);
    } // Just copying over the same recursive formula from the hint
}
```

attendance
bit.ly/abhi-attendance

feedback
bit.ly/abhi-feedback

slides: **bit.ly/abhi-disc**