I believe that people either “get” CS/programming, or they don’t.

A. Yes
B. No
CSE 8B Today

MORE memory models

The Arrays class

Creating simple GUIs

Javadoocs

WIC GBM today, 3pm, B225

PSA3 is posted and ready to go!
(Also: These slides will likely last us through Monday)

Theme for next week: You can do it!
(We’re here to help—but you have to come get it)

START PSA3 EARLY!
It’s ready now

The TAs are lonely
Visit them in office hours!

Seat Requests: See Piazza
Deadline Sat.
1. Colors are made up of which three color components?

A. Red, White, Blue

B. Yellow, Magenta, Cyan

C. Red, Green, Blue

D. Black, White, Gray
2. Suppose you want to set the foreground color of a JButton called jbt. Write a line of Java to change the color to blue.

A. jbt.setForeground(new Color("BLUE"));

B. jbt.setForeground(Color.BLUE);

C. jbt.setForeground(BLUEx);

D. jbt.setForeground("BLUE");
3. The constructor for Font is as follows:
   public Font(String name, int style, int size);
   Give some examples of legal values for the style parameter.

A. Font.BOLD, Font.ITALIC, Font.PLAIN

B. 14, 12, 20

C. “Serif”, “SansSerif”, “Monospaced”
public boolean anyGreaterThanN(int[] myArray, int num) {
    for (int x : myArray) {
        if (x > num) {
            myArray = new int[5];
            for (int i = 0; i < 5; i++)
                myArray[i] = 1;
            return true;
        }
    }
    return false;
}

public static void main(String[] args) {
    ArrayPlay ap = new ArrayPlay();
    int[] myA = {2, 4, 6, 3, 15};
    ap.anyGreaterThanN(myA, 5);
}

What is the value of myA at the end of main?
E. I don’t know
Which is the correct final memory model*?

```csharp
int[] array1 = {1, 2, 3};
int[] array2 = array1;
array2[1] = 5;
array2 = new int[4];
```

* I’m showing arrays that are not referenced by anything, but this is not necessary in a mm.

A. 

B. 

C. 

D. 

*ANSWER: D.*
Which is the correct final memory model*?

int[] array1 = {1, 2, 3};
int[] array2 = array1;
array2[1] = 5;
array2 = new int[4];
The built-in Arrays class has useful methods for manipulating arrays. It is found in the package java.util. You can find its javadoc documentation here: http://docs.oracle.com/javase/6/docs/api/java/util/Arrays.html

```java
static void sort(int[] a)
    Sorts the specified array of ints into ascending numerical order.
```
The built-in Arrays class has useful methods for manipulating arrays. It is found in the package java.util. You can find its javadoc documentation here: http://docs.oracle.com/javase/6/docs/api/java/util/Arrays.html

```
static void sort(int[] a)
    Sorts the specified array of ints into ascending numerical order.
```

You can create javadocs for your code too! Try it!
The Arrays class

The built-in Arrays class has useful methods for manipulating arrays. It is found in the package java.util. You can find its javadoc documentation here: http://docs.oracle.com/javase/6/docs/api/java/util/Arrays.html

```java
static void sort(int[] a)
    Sorts the specified array of ints into ascending numerical order.
```

```java
> int[] intA = {5, 20, 13, 10};
> Arrays.sort( intA );
Static Error: Undefined name 'Arrays'
```

Why did I get this error?
A. sort is not defined for arrays of integers
B. I misspelled the class name “Arrays”
C. I used the wrong syntax for calling the sort method
D. I haven’t included enough information for java to find the Arrays class
E. I don’t know
The Arrays class

The built-in Arrays class has useful methods for manipulating arrays. It is found in the package java.util. You can find its javadoc documentation here: http://docs.oracle.com/javase/6/docs/api/java/util/Arrays.html

```
static void sort(int[] a)
    Sorts the specified array of ints into ascending numerical order.
```

```
> int[] intA = {5, 20, 13, 10};
> java.util.Arrays.sort( intA );

OR

> import java.util.Arrays;
> int[] intA = {5, 20, 13, 10};
> Arrays.sort( intA );

OR

> import java.util.*;
> int[] intA = {5, 20, 13, 10};
> Arrays.sort( intA );
```
The Arrays class

The built-in Arrays class has useful methods for manipulating arrays. It is found in the package java.util. You can find its javadoc documentation here: http://docs.oracle.com/javase/6/docs/api/java/util/Arrays.html

```java
static void sort(int[] a)
    Sorts the specified array of ints into ascending numerical order.

> int[] intA = {5, 20, 13, 10};
> java.util.Arrays.sort( intA );
```

What is the value of the array that intA refers to after the call to sort?

A. {5, 20, 13, 10}
B. {5, 10, 13, 20}
C. {20, 13, 10, 5}
D. {}
E. I don’t know
The built-in Arrays class has useful methods for manipulating arrays. It is found in the package java.util. You can find its javadoc documentation here: http://docs.oracle.com/javase/6/docs/api/java/util/Arrays.html

```java
static void sort(int[] a)
    Sorts the specified array of ints into ascending numerical order.
```

```java
> int[] intA = {5, 20, 13, 10};
> int[] newIntA = java.util.Arrays.sort( intA );
Static Error: Bad types in assignment: from void to int[]
```

Why did I get this error? (Discuss with your group)

The method sort returns void (nothing). You cannot set a variable that is supposed to hold an int array to be equal to void.
Sorting Strings

The built-in Arrays class has useful methods for manipulating arrays. It is found in the package java.util. You can find its javadoc documentation here: http://docs.oracle.com/javase/6/docs/api/java/util/Arrays.html

```java
static void sort( Object[] a )
    // Sorts the specified array of objects into ascending order, according to the
    // natural ordering of its elements.

> String[] stringA = {"Apple", "banana", "Zebra", "mouse"}
> java.util.Arrays.sort( stringA )
```

What is the value of the array that `stringA` refers to after the call to `sort`?

A. {“Apple”, “banana”, “mouse”, “Zebra”}
B. {“banana”, “mouse”, “Apple”, “Zebra”}
C. {“Apple”, “Zebra”, “banana”, “mouse”}
D. {“Apple”, “banana”, “Zebra”, “mouse”}
E. I don’t know

The ASCII/Unicode codes for capital letters are all smaller than those for lowercase letters.
Simple GUIs

In the book you will see a lot of subclasses (extends JFrame). We haven’t learned this yet so don’t worry about it. In our code this week we will just create JFrames and add other components to them. Also, don’t worry too much about this diagram (yet). We will revisit it in about 2-3 weeks, at which point it will all make sense.

(subclasses)
(for now only)
Simple GUIs: Focus on the “green boxes”

- **JFrame**: A top-level window/container to put other graphical components in
- **JLabel**: A component that you can add text to (among other things)
- **JButton**: A component that the user can press
- **FlowLayout, GridLayout, BorderLayout**: Helper classes that govern where components appear in the JFrame (or the JPanel)
- **JPanel**: A component that can store other components (to help you arrange them neatly)

A very rough guide to creating a simple GUI:

1. Make a JFrame
2. Set its properties
3. Add a Layout Manager
4. Create and add components
5. Make the JFrame visible

You don’t always have to do this in this order. Some of the steps can have substeps (e.g., creating a JPanel to organize other components)
```java
public static void main(String[] args) {
    JFrame myFrame = new JFrame("This is my window");
    myFrame.setSize(300, 400);
    myFrame.setVisible(true);
}
```
Simple GUIs

```java
public static void main(String[] args) {
    JFrame myFrame = new JFrame("This is my window");
    myFrame.setSize(300, 400);
    //myFrame.setVisible(true);
}
```
public static void main(String[] args) {
    JFrame myFrame = new JFrame("This is my window");
    myFrame.setSize(300, 400);

    FlowLayout flow = new FlowLayout(FlowLayout.LEFT);
    myFrame.setLayout(flow);
    for (int i = 0; i < 20; i++) {
        JLabel label = new JLabel("CSE 8B");
        myFrame.add(label);
    }

    myFrame.setVisible(true);
}
What is the difference?

FlowLayout flow = new FlowLayout( FlowLayout.LEFT );
myFrame.setLayout( flow );

vs.

myFrame.setLayout( new FlowLayout( FlowLayout.LEFT ) );

Which is a true statement about the two snippets of code above?
A. The first one creates a FlowLayout object, but the second one does not
B. The first one works, but the second one causes an error
C. They invoke a different version of JFrame’s setLayout method
D. They both pass a reference to a new FlowLayout object into JFrame’s setLayout method
E. None of the above
public static void main(String[] args) {
    JFrame myFrame = new JFrame("This is my window");
    myFrame.setSize(300, 400);

    FlowLayout flow = new FlowLayout(FlowLayout.LEFT);
    GridLayout grid = new GridLayout(5, 6);
    myFrame.setLayout(grid);
    for (int i = 0; i < 20; i++) {
        JLabel label = new JLabel("CSE 8B");
        myFrame.add(label);
    }

    myFrame.setVisible(true);
}
public static void main(String[] args) {

    JFrame myFrame = new JFrame("This is my window");
    myFrame.setSize(300, 400);

    FlowLayout flow = new FlowLayout(FlowLayout.LEFT);
    GridLayout grid = new GridLayout(5, 6);
    myFrame.setLayout(flow);
    for (int i = 0; i < 20; i++) {
        JLabel label = new JLabel("CSE 8B");
        label.setFont(new Font("Serif", Font.ITALIC, i*3));
        myFrame.add(label);
    }
}

Sketch (more or less) what you think this GUI will look like