CSE [8, 8, 8, 8]B Today

PSA 1: fourInARow
Due Monday 11:59pm – TUTOR HOURS START TOMORROW!

return, arrays and methods

String and char
PSAs: rules for working with a partner

• PSA1: INDIVIDUAL → IT WILL BE MARKED with [I] on the web
• PSA2: YOU CHOOSE

• Please do! (Again, remember 1000s of students, only 100s of lab machines—you do the math)
• You must PAIR PROGRAM
  – You must always be together when coding
  – One person drives, one person navigates
• **You must include in your header comment:**
  – Both partners’ names
  – Both partners’ PIDs
  – Both partners’ cs8b logins
    (cs8sb... look for it following instructions in PSA1)
The ArrayPlay class

The following CODE is in a class called ArrayPlay. ALMOST identical to the class you will work with in PSA1 (Except it has a different name so that I can provide you with the ArrayPlay code separately from the assignment starter code. But the examples should be very similar to your assignment.)
The ArrayPlay class

/** ArrayPlay is a class that implements some basic matching functions on a list */

public class ArrayPlay
{
    private int[] storedArray; // FIELD!

    public ArrayPlay( int[] inputArray ) // CONSTRUCTOR!!
    {
        storedArray = new int[inputArray.length];
        for ( int i = 0; i < inputArray.length; i++ )
        {
            storedArray[i] = inputArray[i];
        }
    }

    public static void main( String[] args )
    {
        int[] a = {1, 2, 3};
        ArrayPlay am = new ArrayPlay( a );
        a[0] = 42;
        System.out.println( am.storedArray[0] );
    }

    What does the last line of main print?   A. 0   B. 1   C. 42   D. Other

Notice that the constructor of this method copies the items out of the inputArray into the storedArray.
/** greaterThanN returns true if the element at position index in storedArray is greater than num.

* @param num The N to compare against
* @param index The index of the element in question
* @return true if storedArray[index] is greater than num, false otherwise
* */

public ____A____ greaterThanN( ____B____ index, int num )
{
    if ( ____C____ )
        return true;
    else
        return false;
}
Complete the following method which returns true if any element in storedArray is greater than num, and false if no element is greater than num

```java
public boolean anyGreaterThanN(int num) {
    for (int x : storedArray) {
        // Your code here
    }
}
```
Summary so far ...

- Computation takes place (generally) by calling methods
- Methods can return values and modify the data passed to them
- Parameters are passed by value to methods
- Primitives and objects/arrays are represented differently in memory (box-and-arrows)
- There are often many ways to solve the same problem.

Tips for success on PSA1

- Compile and run A LOT
- Work in the lab
- Trace your code on paper (no “debugging by random perturbation”)
- Think through your approach before you code
- Ask for help, don’t waste your time if you are really stuck.

(if you get really stuck with PSA1, I’ll go to TAs or instructor office hours ASAP!)

PSA1 has to be done INDIVIDUALLY! (other PSAs you’ll choose)
PSA2 – processing strings

“Strings” and ‘c’ ‘h’ ‘a’ ‘r’ ‘s’ in Java

Implementing your very own Secret Decoder Ring in Java...

... and a little Scrabble to warm up

PSA2 Will be posted TUESDAY.
START EARLY! It’s MUCH harder than PSA1
Chars and ints: deceptively similar

What is the value of each of the expressions below:

Remember, casting a value changes its type

(1) > (int)'A'
(2) > 'A' + 2
(3) > (char)('A' + 2)

A. error, error, error
B. 65, error, error
C. 65, ‘C’, ‘C’
D. 65, 67, ‘C’
public class StringPlay {
    public boolean hasLetter(String word, char letter) {
        for (int i = 0; i < word.length(); i++) {
            if (word.charAt(i) == letter) {
                return true;
            }
        }
        return false;
    }
}

StringPlay s = new StringPlay();
System.out.println(s.hasLetter("Sleep", 'S'));
public class StringPlay {
    public boolean hasLetter( String word, char letter ) {
        for ( int i = 0; i < word.length(); i++ ) {
            if ( word.charAt( i ) == letter ) {
                return true;
            }
        }
        return false;
    }
}

StringPlay s = new StringPlay();
System.out.println( s.hasLetter( "Sleep", 's' ) );
public class StringPlay {
    public boolean hasLetter( String word, char letter ) {
        for ( int i = 0; i < word.length(); i++ ) {
            if ( word.charAt( i ) == letter ) {
                return true;
            }
        }
        return false;
    }
}

StringPlay s = new StringPlay();
System.out.println( s.hasLetter( "Sleep", 's' ) );

If we want this to return true, how could we ignore case?
public class StringPlay {
    public boolean hasLetter( String word, char letter ) {
        letter = Character.toLowerCase( letter );
        for ( int i = 0; i < word.length(); i++ ) {
            char current = Character.toLowerCase( word.charAt( i ) );
            if ( current == letter ) {
                return true;
            }
        }
        return false;
    }
}
public class StringPlay {
    public boolean hasLetter( String word, char letter ) {
        letter = Character.toLowerCase( letter );
        for ( int i = 0; i < word.length(); i++ ) {
            char current = Character.toLowerCase( word.charAt( i ) );
            if ( current == letter ) {
                return true;
            }
        }
        return false;
    }
}

> StringPlay s = new StringPlay();
> String check = new String( "Sleep" );
> s.hasLetter( check, 's' );
> System.out.println( check );

What will be printed?
A. “Sleep”
B. “sleep”
C. “true”
D. “false”
E. I don’t know
Creating new Strings (worksheet)

Complete the following method to replace all instances of one character with another in a string (and the return the resulting string). Your method should be case sensitive.

\[ \text{e.g., } \text{replaceChar}( \text{"One for the money, YO!"}, 'O', 'i' ) \rightarrow \text{"ine for the money, Yi!"} \]

```java
public String replaceChar( String s, char gone, char here )
{
    char[] chars = s.toCharArray();
    return String.valueOf( chars ); // a new String of chars
}
```