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

PSA 1: fourInARow
Due Thursday 11:59pm

Tutor Hours

The calendar below lists the open lab tutoring hours for the term. It should always be current. Hours may be staffed (and no one will be there, so you probably don’t want to show up). Discussion sections and office hours below. All tutor hours are in CSE 8260 unless otherwise indicated.

CSE 8b Tutor Hours Spring 2013

Tutor: TBA
Tutor: Donna N.
Tutor: Pavel, Lisa

return, arrays and methods, oh my!

I sure could use some help
1. You've been very busy with work and your other classes and you've gotten a little behind on your CSE 8b PSA. You've so far submitted all of your 8b PSAs on time. What should you do to get an extension on this one?

A. Email the professor and explain the situation, asking for an extension
B. Email the professor and ask if you can use a slip day
C. Post on Piazza to ask if you can use a slip day
D. Turn the PSA in up to 24 hours late and you will automatically be charged a slip day
E. Turn in the homework before the deadline, no matter what, or you will get a 0.
2. Does forgetting to bring your iClicker to class count as a missed class?

A. Yes, it counts as a missed class

B. No, it does not count as a missed class

C. It only counts as a missed class if you do not let the professor know that you forgot your iClicker
3. What should you do to request a regrade on an assignment?

A. Email the person who graded your assignment. If you don’t have their email, post a message with their name/login on Piazza (can be private)

B. Email the professor, always

C. Wait until the end of the quarter and then ask for all of your regrades at once

D. Nothing, there are no regrades in CSE 8B
4. What are two ways you can fail CSE8b, even if your average grade is above 60% (or even 70 or 80%)?

A. Missing more than one PSA; missing more than one exam

B. Missing more than 8 class sessions; getting lower than a 55% on the final

C. Missing more than 6 class sessions; having lower than a 55% average on the PSAs

D. Having lower than a 55% average on the reading quizzes; missing more than 8 class sessions

E. Posting too many questions on Piazza; coming to office/lab hours too much
5. Why can't you use the assignment operator (=) to make a copy of an array, i.e. array1 = array2?

A. The assignment operator only copies the reference to array2, so array1 and array2 still point to the same underlying data.

B. The assignment operator is not defined for arrays.

C. You can. There’s no problem with doing this.
6. Assume you have a 3x3 array of ints stored in the variable myArray. All positions in the array are set to 0. What are the values of myArray[1] and myArray[1].length?

A. myArray[1] is 0; myArray[1].length is 0

B. myArray[1] is 3; myArray[1].length is 3

C. myArray[1] is [0,0,0]; myArray[1].length is 3

D. myArray[1] is [0]; myArray[1].length is 1

E. myArray[1] is 1; myArray[1].length is 1
Methods: the Heart of Computation

Javadoc comments! Use them!

/**
* greaterThanN returns true if the element at position index
* is greater than N.
* @param myArray An array of integers
* @param num The N to compare against
* @param index The index of the element in question
* @return true if myArray[index] is greater than num, false otherwise
* */
public ____B____ greaterThanN( ____B____ myArray, int index, int num )
{
    if ( ____C____ )
        return true;
    else
        return false;
}
/** greaterThanN returns true if the element at position index
 * is greater than N.
 * @param myArray An array of integers
 * @param num The N to compare against
 * @param index The index of the element in question
 * @return true if myArray[index] is greater than num, false otherwise
 */

public boolean greaterThanN( __(B)__ myArray, int index, int num )
{
    if ( ______(C)_______________ )
        return true;
    else
        return false;
}
Methods: the Heart of Computation

Javadoc comments! Use them!

/** greaterThanN returns true if the element at position index
 * is greater than N.
 * @param myArray An array of integers
 * @param num The N to compare against
 * @param index The index of the element in question
 * @return true if myArray[index] is greater than num, false otherwise
 * */

public boolean greaterThanN(int[] myArray, int index, int num) {
    if (________(C)_____________ )
        return true;
    else
        return false;
}
Javadoc comments! Use them!

/** greaterThanN returns true if the element at position index * is greater than N.
 * @param myArray An array of integers
 * @param num The N to compare against
 * @param index The index of the element in question
 * @return true if myArray[index] is greater than num, false otherwise
 * */

public boolean greaterThanN(int[] myArray, int index, int num) {
    if (myArray[index] > num)
        return true;
    else
        return false;
}
public class ArrayPlay {
    public boolean greaterThanN( int[] myArray, int index, int num )
    {
        if ( myArray[index] > num )
            return true;
        else
            return false;
    }

    public static void main( String[] args )
    {
        ArrayPlay ap = new ArrayPlay();
        int[] myA = {2, 4, 10, 1};
        System.out.println( ap.greaterThanN( myA, 2, 6 ) );
    }
}

- All methods must be defined in a class
- To call them method, you must have an object (unless the method is static)
- There is a special method (main) that gets called automatically
Java: It’s about classes (and objects)

```java
public class ArrayPlay {
    public boolean greaterThanN( int[] myArray, int index, int num ) {
        if ( myArray[index] > num )
            return true;
        else
            return false;
    }

    public static void main( String[] args ) {
        ArrayPlay ap = new ArrayPlay();
        int[] myA = {2, 4, 10, 1};
        System.out.println( ap.greaterThanN( myA, 2, 6 ) );
    }
}
```

What will this method print?

A. true
B. false
C. It will cause an error
D. I don’t know
A slight change, and...

public class ArrayPlay {
    public boolean greaterThanN( int[] myArray, int index, int num )
    {
        if ( myArray[index] > num )
            return true;
        else
            return false;
    }
    public static void main( String[] args )
    {
        ArrayPlay ap = new ArrayPlay();
        int[] myA = {2, 4, 10, 1};
        System.out.println( ap.greaterThanN( myA, 6, 2 ) );
    }
}

What will this method print?
A. true
B. false
C. It will cause an error
D. I don’t know
public class ArrayPlay
    public boolean greaterThanN( int[] myArray, int index, int num )
    {
        if ( _____________________________  )
            return false;
        else
            return true;
    }

What goes in the blank so that this method does the same thing as before?
A. myArray[index] > num
B. myArray[index] >= num
C. myArray[index] < num
D. myArray[index] <= num
E. I don’t know

HINT: Try it with the values [2, 3, 4], 1 and 3 (should return false because 3 is NOT greater than 3).
Getting loopy... worksheet!

Complete the following method which returns true if any element in myArray is greater than num, and false if no element is greater than num.

```java
public boolean anyGreaterThanN( int[] myArray, int num )
{
    for ( int x : myArray )
    {
        for ( int x : myArray )
        {
            if ( x > num ) {
                return true;
            } // end if
        } // end for
    } // end for
    return false;
}
```

Hey, it's a for-each loop, where x takes the value of each element in the array. 

```java
[1, 5, 2] num = 4
```
Getting loopy... worksheet!

Complete the following method which returns true if any element in myArray is greater than num, and false if no element is greater than num:

```java
public boolean anyGreaterThanN( int[] myArray, int num ) {
    for ( int x : myArray )
    {
        if ( x > num ) {
            return true;
        } // end if
        else {
            return false;
        } // end else
    } // end for
    return false;
}
```

Which of the following correctly completes the method?

A.
```java
for ( int x : myArray )
{
    if ( x > num ) {
        return true;
    } // end if
} // end for
else {
    return false;
} // end else
```

B.
```java
for ( int x : myArray )
{
    if ( x > num ) {
        return true;
    } // end if
} // end for
```

C.
```java
for ( int x : myArray )
{
    if ( x > num ) {
        return true;
    } // end if
}
} // end for
return false;
```
Summary

• 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
• As for help, don’t waste your time if you are really stuck.