Extra lab hours today
Check the calendar –
and be sure to thank your tutors!

Do you plan to take another CSE course?
A. Yes, definitely
B. Probably
C. Maybe
D. Probably not
E. No, definitely not

Final exam prep:
• Review today in discussion
• Review sheet posted (see Piazza)
• My office hours (Tuesday 9-11am)
• TA office hours finals week (see Piazza/course calendar)
• BUT, the MOST important thing: REDO CLICKER QUESTIONS, WORKSHEETS and PSAs!!
public int binarySearch( ArrayList<Integer> list,  
        int toFind, int low, int high ) {
    int mid = (low+high) / 2;
    if (toFind < list.get( mid ))
        return binarySearch(list, toFind, low, mid-1);  // search left
    else if (toFind == list.get( mid ))
        return mid;
    else // toFind is > list.get(mid)
        return binarySearch(list, toFind, mid+1, high);  // search right
}
public int binarySearch( ArrayList<Integer> list, int toFind, int low, int high ) {
    int mid = (low+high) / 2;
    if (toFind < list.get(mid))
        return binarySearch( list, toFind, low, mid-1 );
    else if (toFind == list.get(mid))
        return mid;
    else // toFind is > list.get(mid)
        return binarySearch( list, toFind, mid+1, high );
}

Which is the base case in the method above?

A. if (toFind < list.get(mid))
B. return binarySearch( list, toFind, low, mid-1 );
C. else if (toFind == list.get(mid))
D. return binarySearch( list, toFind, mid+1, high );
public int binarySearch( ArrayList<Integer> list, int toFind, int low, int high ) {

    int mid = (low+high) / 2;
    if (toFind < list.get( mid ))

        return binarySearch( list, toFind, low, mid-1 );

    else if (toFind == list.get( mid ))

        return mid;

    else // toFind is > list.get(mid)

        return binarySearch( list, toFind, mid+1, high );
}

When will this method not work?
A. When the element you are looking for is the first element in the list
B. When the element you are looking for is the last element in the list
C. When the element you are looking for is not in the list
D. When the list is not sorted.
E. More than one of these (C and D).
The CSE 8B central skills

• Write
  • Programs/algorithms
  • Using loops, arrays, conditionals, variables objects, subclasses, GUIs, recursion, etc (see review sheet)

• Trace/debug
  • Draw memory models
  • Given some code, what does it do?
  • Find the errors and fix them

• Explain
  • Why things go wrong
  • How things work/why code functions the way it does
  • Why the answer to a question is right or wrong

You will be given reference material. Memorize concepts not details.
Write: code that solves an array problem

Write a method that takes an array of ints and returns a new array of ints that contains only a single instance of each of the numbers in the input array. You may assume you have the following helper method:

```java
public int countUnique(int[] a) {
    // Implementation
}
```

Which will return the number of unique elements in `a`. Hint: consider a helper method that will check whether an array contains a given element.

```java
public class ArrayPlay {
    public int[] removeDuplicates(int[] arr) {
        int[] toReturn = new int[countUnique(arr)];
        int retInd = 0;
        for (int i = 0; i < arr.length; i++) {
            if (!contains(toReturn, arr[i])) {
                toReturn[retInd] = arr[i];
                retInd++;
            }
        }
        return toReturn;
    }
}
```

- `contains(int[] a, int elem)` returns `true` if `elem` is in `a`.

Note: The provided solution is illustrative and may need adjustments for correctness.
public class ArrayPlay
    public int[] removeDuplicates( int[] arr )
    {
        int[] myArray = new int[countUnique( arr )];
        int retInd = 0;
        for ( int i = 0; i < arr.length; i++ )
        {
            if ( !contains( myArray, arr[i] ) ) {
                myArray[retInd] = arr[i];
                arr[i] = 0;
            }
        }
        return myArray;
    }
    public static void main( String[] args )
    {
        ArrayPlay ap = new ArrayPlay();
        int[] myArray = {1, 1, 2, 5, 2, 2, 3};
        int[] myArrayNoDups = ap.removeDuplicates( myArray );
    }

What is the value of myArray in main at the end of main (assume contains works correctly, as in the previous code)? What is the value of myArrayNoDups?

Value of myArray:
A. [1, 1, 2, 5, 2, 2, 3]
B. [1, 2, 5, 3]
C. [0, 1, 0, 0, 0, 2, 0]
D. [0, 1, 0, 0, 2, 2, 0]
E. [1, 0, 2, 5, 0, 0, 3]
public class ArrayPlay
    public int[] removeDuplicates( int[] arr )
    {
        int[] myArray = new int[countUnique( arr )];
        int retInd = 0;
        for ( int i = 0; i < arr.length; i++ )
        {
            if ( !contains( myArray, arr[i] ) ) {
                myArray[retInd] = arr[i];
                arr[i] = 0;
            }
        }
        return myArray;
    }

public static void main( String[] args )
{
    ArrayPlay ap = new ArrayPlay();
    int[] myArray = {1, 1, 2, 5, 2, 2, 3};
    int[] myArrayNoDups = ap.removeDuplicates( myArray );
}
class Person
{
    public String getName() {...}
    // no getMajor method defined here
}

class Student extends Person
{
    // no getName defined here
    public String getMajor() {...}
}

class General extends Person
{
    // no getName defined here
    public String getMajor() {...}
}

Person P = new Person();
Person Q = new Person();
Student S = new Student();
General G = new General();
Object Ob = new General();

Which of the five lines of code below will work as written and which will Java complain about (and why?)

Is any casting needed?

1. print(S.getName());
2. P = S;
3. print(P.getMajor());
4. G = Q;
5. G = Ob;

This code was previously posted on the schedule and assignments page. You can run it for yourself.
public class MorePeople {
    public static void main( String[] args )
    {
        Person p = new Person( "Sally" );
        Person s = new Student( "Sam" );
        s.setName( "Steve" );
        System.out.println( p.getName() );
        System.out.println( s.getName() );
    }
}

class Person {
    private String name;
    public Person( String n ) { name = n; }
    public String getName() { return this.name; }
}

class Student extends Person  
{
    public Person( String n ) { name = n; }
    public void setName( String newName )
    { name = newName; }
}