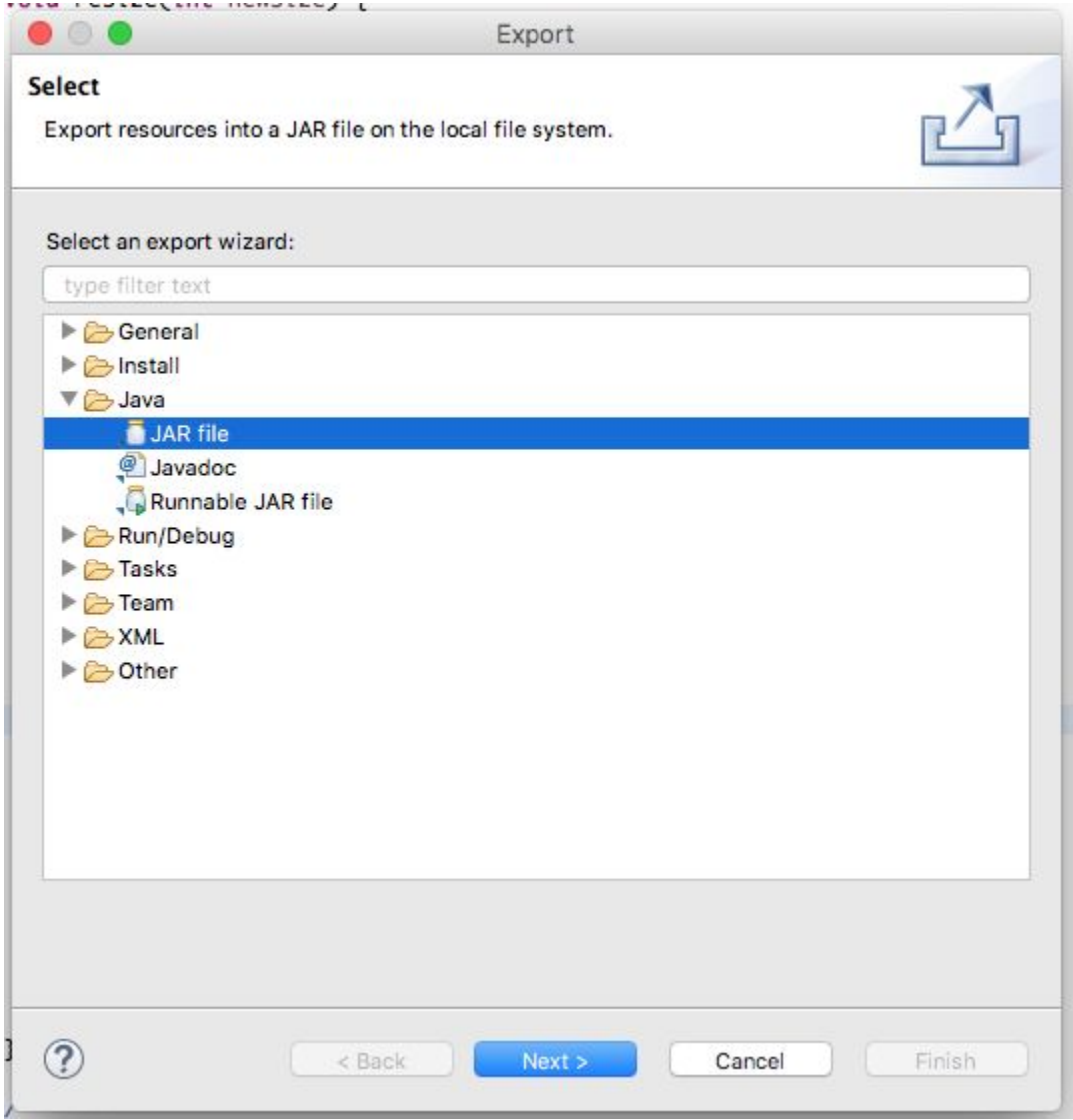
ICS-211 Lab Assignment 1

Today...

- Updated source code on GitHub from Wed. with bugs fixed and additional test case
- Introduce Assignment 1
- Work on assignment 1 (due Sept. 9)
- These slides are at https://github.com/psoulier/CircularBuff211

Submitting Assignment

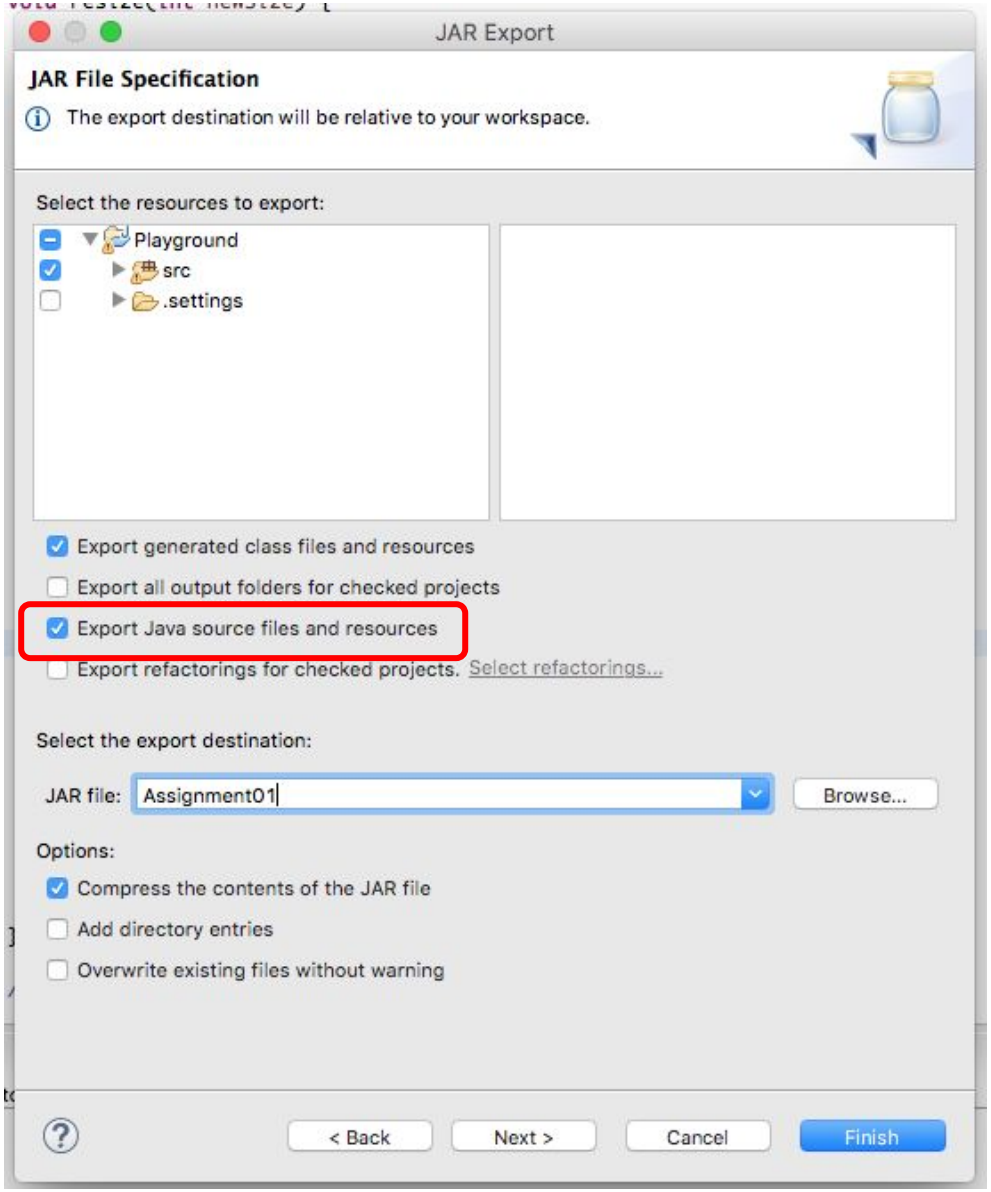
- Submit to Laulima
- Export your project using: File | Export
- Pick “JAR File”



Submitting Assignment

- Be sure to select the export source files option!!!
- Put your README file in your project so it's included
- README must be either a text file or PDF.
- Check the contents of your JAR file:

```
> jar tf Assignment01.jar
```



Assignment #1 - What you'll be doing...

```
class ArraySort {
     public void insertionSort(E[] data, Comparator<? Super E> compare) {}
     public void bubbleSort(E[] data, Comparator<? Super E> compare) {}
     public void selectionSort(E[] data, Comparator<? Super E> compare) {}

     // Returns the time it took to sort array.
     public long getSortTime() {}

     // Returns the number of comparisons needed to sort array.
     public long getCompareCount() {}

     // Return number of swaps used to sort array
     public long get getSwapCount() {}
}
```

- Use the method names *exactly* as specified above

Comparators

- Provides mechanism to compare two objects of the same type
- Allows your sort methods to sort an array of *any* type of object
- `Comparator` is an interface with two methods:
  - `int compare(E a, E b)`
  - `boolean equals(Object obj)`
  - You only need to implement the `compare` method (we'll ignore `equals` for now)
- The `compare` method returns:
  - -1 if a < b
  - 0 if a == b
  - 1 if a > b

Comparators - Simple Example

```
class CompareNumbers implements Comparator<Integer> {
     public int compare(Number a, Number b) {
          return a.compareTo(b);
     }
}


Integer[]     numbers = {1, 4, -2, 9, 113};
ArraySort     arrSort = new ArraySort();

arrSort.bubbleSort(numbers, new CompareNumbers());
```

Comparators - More Interesting Example

```
class Student {
     public int    age;
     public double gpa;
}

class CompareAge implements Comparator<Student> {
     public int compare(Student a, Student b) {
          if (a.age < b.age) { return -1; }
          else if (a.age == b.age) { return 0; }
          else { return 1; }
     }
}


Student[]     students = new Students[100];
ArraySort     arrSort = new ArraySort();

// Code to populate “students” array...

arrSort.bubbleSort(students, new CompareAge());
```