EECS2011 A Assignment 1 Solutions Fall 2017

Our solution directory **A1sol** (also the package name) includes all solution files and can be found here . The file you are reading is **a1sol.pdf** in that directory. The java codes are best viewed by Eclipse.

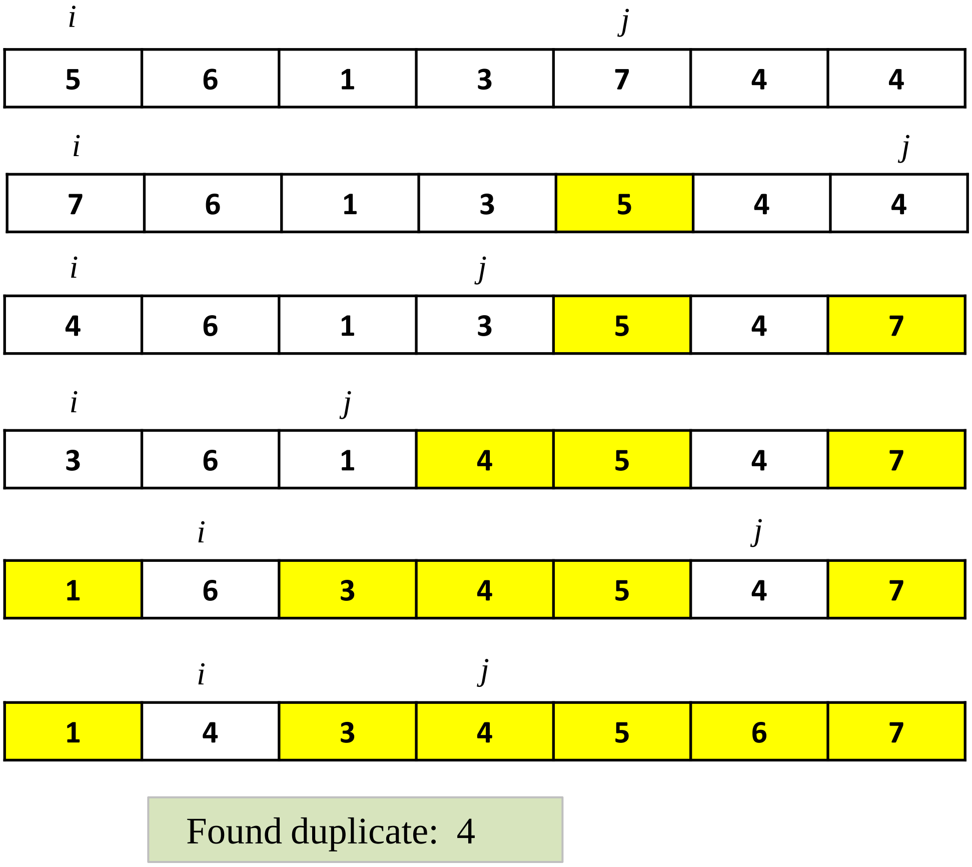
Note: **A1.TestHelper** is imported in the programs for problems 1 and 2.

Problem 1. [32%] Finding a duplicate in a preconditioned array

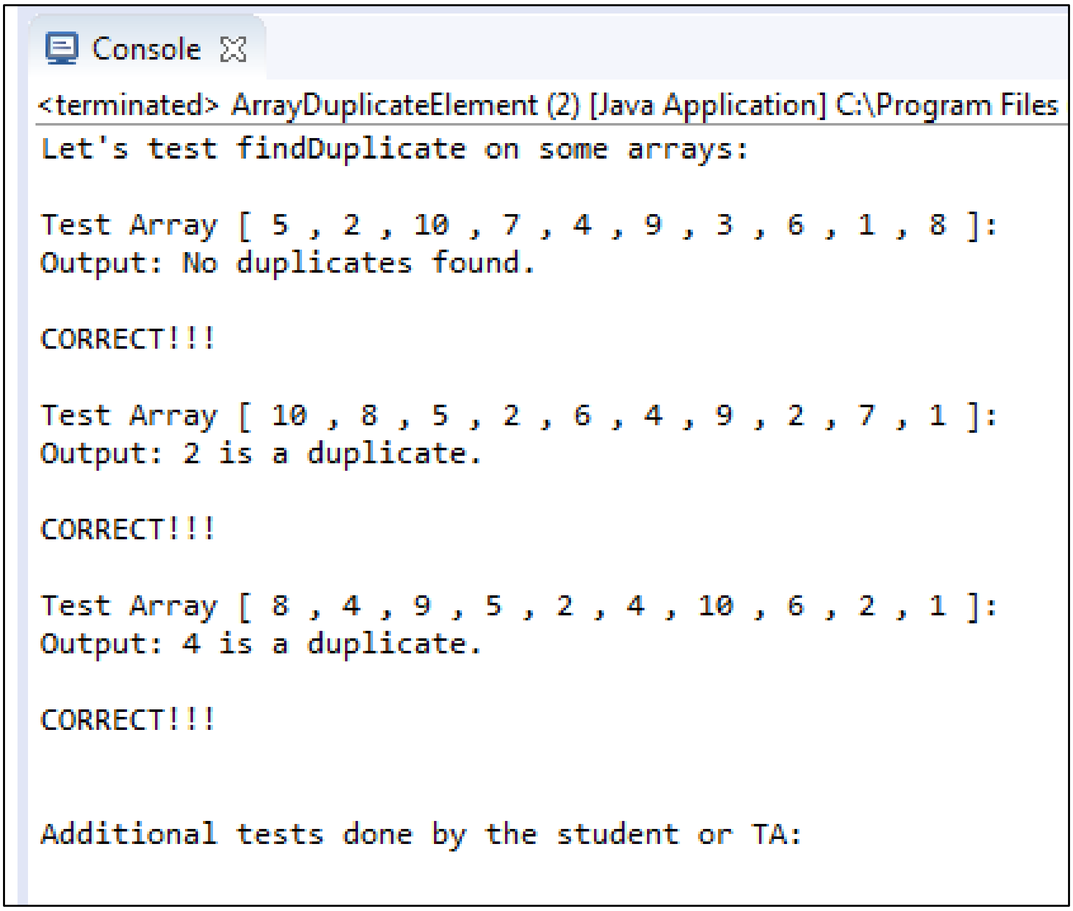
Java Code: *ArrayDuplicateElement.java*

This code gives detailed comments that include: a warning against violating the stated time & space restrictions; explains how the method works & why it takes linear time and is in-place.

An illustrative example: In the figure below index variables i and j used in the code show how the algorithm proceeds after each iteration of the inner while-loop. When a perfect position is found, its entry is coloured yellow.



Test I/O results on the console:



```
<terminated> ArrayDuplicateElement (2) [Java Application] C:\Program Files
Let's test findDuplicate on some arrays:

Test Array [ 5 , 2 , 10 , 7 , 4 , 9 , 3 , 6 , 1 , 8 ]:
Output: No duplicates found.

CORRECT!!!

Test Array [ 10 , 8 , 5 , 2 , 6 , 4 , 9 , 2 , 7 , 1 ]:
Output: 2 is a duplicate.

CORRECT!!!

Test Array [ 8 , 4 , 9 , 5 , 2 , 4 , 10 , 6 , 2 , 1 ]:
Output: 4 is a duplicate.

CORRECT!!!


Additional tests done by the student or TA:
```

Exercise:

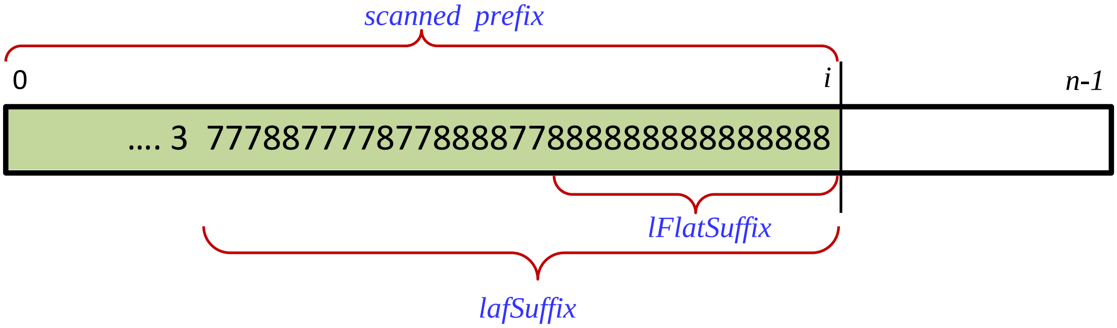
Solve the same problem with the same specifications except with the following slightly modified precondition: each element of the input array is an integer between 1 and 2n, where n denotes the length of the array.

Problem 2. [36%] **Longest Almost Flat Subarray**

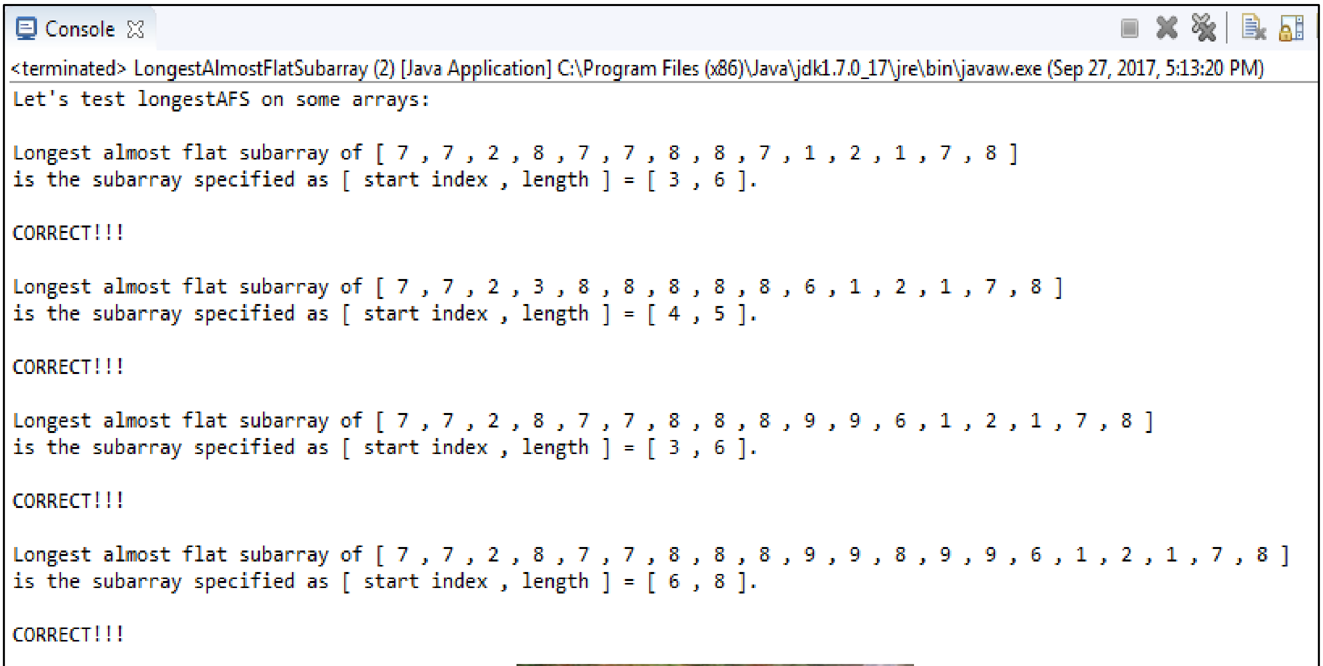
Java Code: *LongestAlmostFlatSubarray.java*

This code gives detailed comments that include: how & why the method works; why it takes linear time and is in-place.

An illustrative example: The figure below illustrates the two subarrays used in the code: *lafSuffix* (longest almost flat suffix) and *lFlatSuffix* (longest flat suffix) of the scanned prefix. What happens if the next scanned element at index $i+1$ is: $7, 8, 9$, or other? Now check to see how the code handles such cases.



Test I/O results on the console:



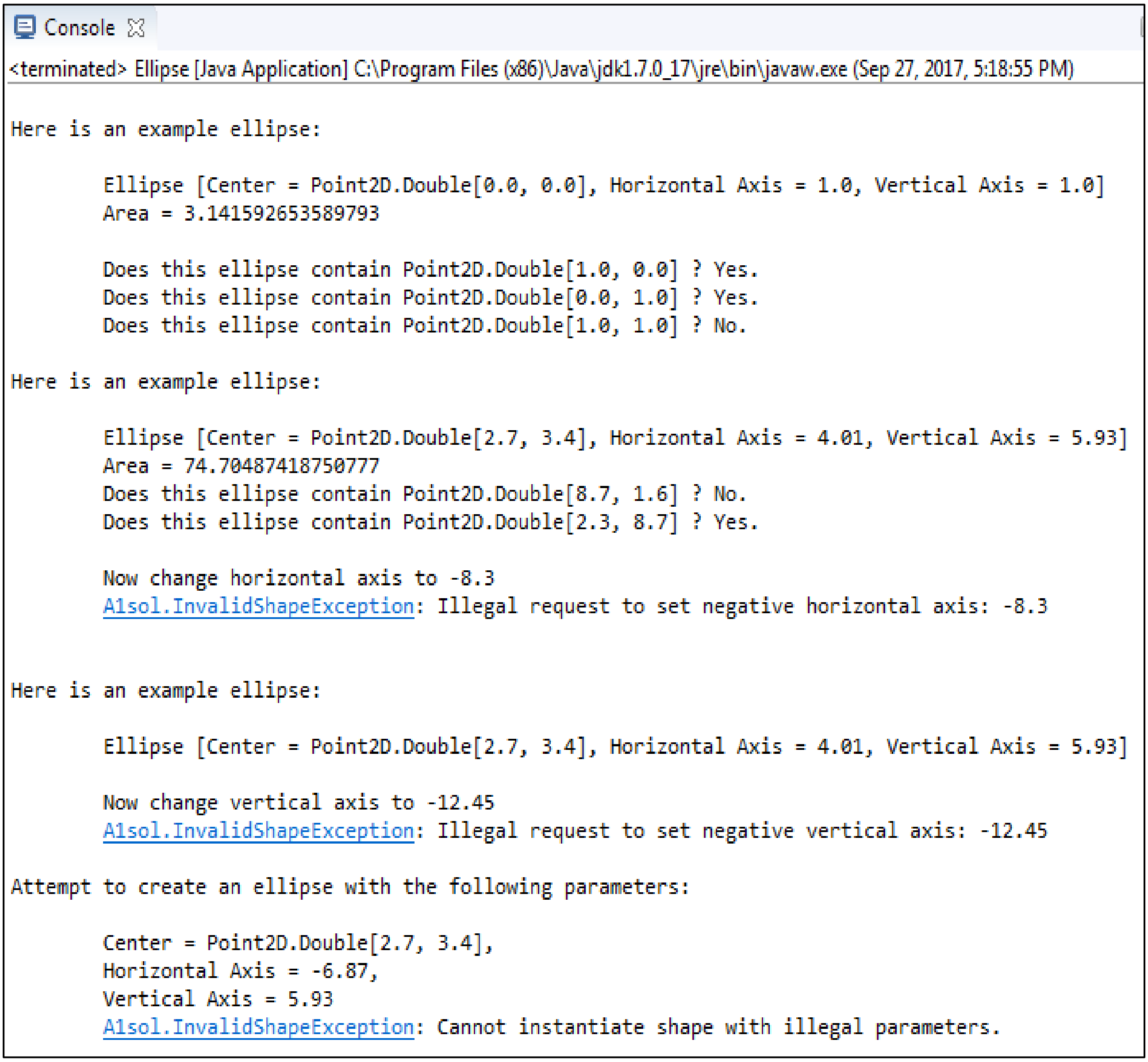
```
<terminated> LongestAlmostFlatSubarray (2) [Java Application] C:\Program Files (x86)\Java\jdk1.7.0_17\jre\bin\javaw.exe (Sep 27, 2017, 5:13:20 PM)
Let's test longestAFS on some arrays:

Longest almost flat subarray of [ 7 , 7 , 2 , 8 , 7 , 7 , 8 , 8 , 7 , 1 , 2 , 1 , 7 , 8 ]
is the subarray specified as [ start index , length ] = [ 3 , 6 ].

CORRECT!!!

Longest almost flat subarray of [ 7 , 7 , 2 , 3 , 8 , 8 , 8 , 8 , 8 , 6 , 1 , 2 , 1 , 7 , 8 ]
is the subarray specified as [ start index , length ] = [ 4 , 5 ].

CORRECT!!!

Longest almost flat subarray of [ 7 , 7 , 2 , 8 , 7 , 7 , 8 , 8 , 8 , 9 , 9 , 6 , 1 , 2 , 1 , 7 , 8 ]
is the subarray specified as [ start index , length ] = [ 3 , 6 ].

CORRECT!!!

Longest almost flat subarray of [ 7 , 7 , 2 , 8 , 7 , 7 , 8 , 8 , 8 , 9 , 9 , 8 , 9 , 9 , 6 , 1 , 2 , 1 , 7 , 8 ]
is the subarray specified as [ start index , length ] = [ 6 , 8 ].

CORRECT!!!


Additional tests done by the student or TA:
```

Problem 3. [32%] A Hierarchy of Planar Shapes

Java Codes: *PlanarShape.java , Ellipse.java, Circle.java, InvalidShapeException.java*

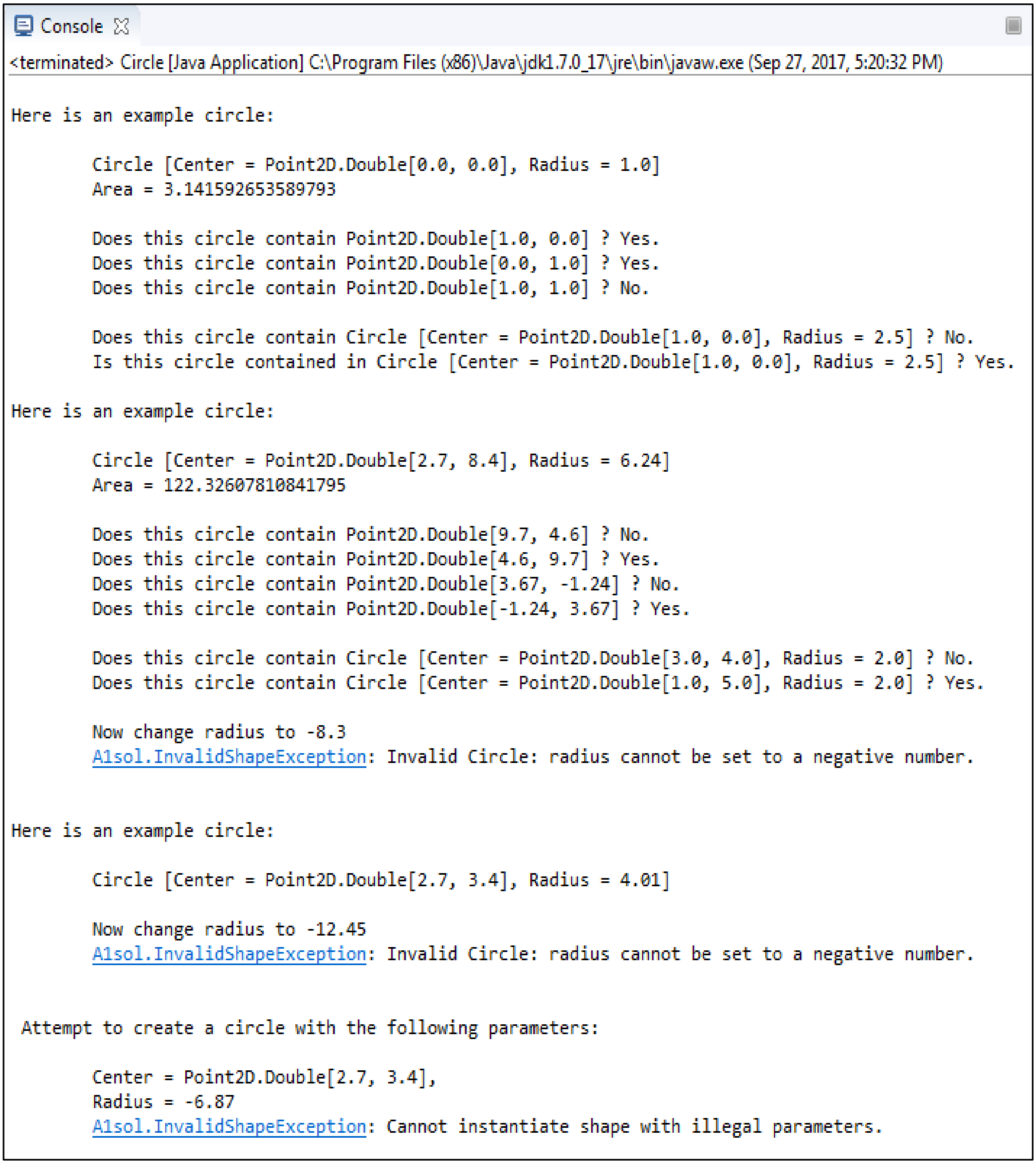
Design Idea: The commented codes should be self explanatory.

Test I/O results on Ellipse:



```
<terminated> Ellipse [Java Application] C:\Program Files (x86)\Java\jdk1.7.0_17\jre\bin\javaw.exe (Sep 27, 2017, 5:18:55 PM)

Here is an example ellipse:

        Ellipse [Center = Point2D.Double[0.0, 0.0], Horizontal Axis = 1.0, Vertical Axis = 1.0]
        Area = 3.141592653589793

        Does this ellipse contain Point2D.Double[1.0, 0.0] ? Yes.
        Does this ellipse contain Point2D.Double[0.0, 1.0] ? Yes.
        Does this ellipse contain Point2D.Double[1.0, 1.0] ? No.

Here is an example ellipse:

        Ellipse [Center = Point2D.Double[2.7, 3.4], Horizontal Axis = 4.01, Vertical Axis = 5.93]
        Area = 74.70487418750777
        Does this ellipse contain Point2D.Double[8.7, 1.6] ? No.
        Does this ellipse contain Point2D.Double[2.3, 8.7] ? Yes.

        Now change horizontal axis to -8.3
        A1sol.InvalidShapeException: Illegal request to set negative horizontal axis: -8.3


Here is an example ellipse:

        Ellipse [Center = Point2D.Double[2.7, 3.4], Horizontal Axis = 4.01, Vertical Axis = 5.93]

        Now change vertical axis to -12.45
        A1sol.InvalidShapeException: Illegal request to set negative vertical axis: -12.45

Attempt to create an ellipse with the following parameters:

        Center = Point2D.Double[2.7, 3.4],
        Horizontal Axis = -6.87,
        Vertical Axis = 5.93
        A1sol.InvalidShapeException: Cannot instantiate shape with illegal parameters.
```

Test I/O results on Circle:



```
<terminated> Circle [Java Application] C:\Program Files (x86)\Java\jdk1.7.0_17\jre\bin\javaw.exe (Sep 27, 2017, 5:20:32 PM)

Here is an example circle:

        Circle [Center = Point2D.Double[0.0, 0.0], Radius = 1.0]
        Area = 3.141592653589793

        Does this circle contain Point2D.Double[1.0, 0.0] ? Yes.
        Does this circle contain Point2D.Double[0.0, 1.0] ? Yes.
        Does this circle contain Point2D.Double[1.0, 1.0] ? No.

        Does this circle contain Circle [Center = Point2D.Double[1.0, 0.0], Radius = 2.5] ? No.
        Is this circle contained in Circle [Center = Point2D.Double[1.0, 0.0], Radius = 2.5] ? Yes.

Here is an example circle:

        Circle [Center = Point2D.Double[2.7, 8.4], Radius = 6.24]
        Area = 122.32607810841795

        Does this circle contain Point2D.Double[9.7, 4.6] ? No.
        Does this circle contain Point2D.Double[4.6, 9.7] ? Yes.
        Does this circle contain Point2D.Double[3.67, -1.24] ? No.
        Does this circle contain Point2D.Double[-1.24, 3.67] ? Yes.

        Does this circle contain Circle [Center = Point2D.Double[3.0, 4.0], Radius = 2.0] ? No.
        Does this circle contain Circle [Center = Point2D.Double[1.0, 5.0], Radius = 2.0] ? Yes.

        Now change radius to -8.3
        A1sol.InvalidShapeException: Invalid Circle: radius cannot be set to a negative number.


Here is an example circle:

        Circle [Center = Point2D.Double[2.7, 3.4], Radius = 4.01]

        Now change radius to -12.45
        A1sol.InvalidShapeException: Invalid Circle: radius cannot be set to a negative number.


 Attempt to create a circle with the following parameters:

        Center = Point2D.Double[2.7, 3.4],
        Radius = -6.87
        A1sol.InvalidShapeException: Cannot instantiate shape with illegal parameters.
```