**Ariel University, School of Computer Science, 2022**

**Introduction to Computer Science**

**Ex4: Main OOP Assignment**

**Abstract**:

This assignment includes a relatively complex set of classes and interfaces. You are asked to implement a set of geometric shapes, a shape container, a basic GUI, and basic save/load capabilities. Moreover, in this assignment, you are required to implement a JUnit class - to test all your functions (Geo, Collections, save & load. GUI classes are not required to have JUNIT tests).

**To Do**

1. Download [Ex4\_V0.1.zip](https://github.com/benmoshe/Intro2CS/blob/main/src/Exe/Ex4_V0.1.zip), and uncompress it.
2. Run the Ex4\_sol-out.jar file (java -jar Ex4\_sol.V0.1-out.jar). This is a complete solution to Ex4 - you should implement your solution accordingly. Try loading the a0 file - you should see an image as shown in Figure 1 (Left and Right). The a2 file is shown in Figure 2.
3. Create a new project (named Ex4), and download all the classes and interfaces of [Ex4](https://github.com/benmoshe/Intro2CS/tree/main/src/Exe/Ex4). These files contain most of the required classes (and all the interfaces). The supplied “skeleton” is runnable (run the Ex4Main t2() function). You should get a simple GUI which will allow you to draw circles.
4. Implement and update your solution to all the [GeoShapeable](https://github.com/benmoshe/Intro2CS/blob/main/src/Exe/Ex4/geo/GeoShapeable.java) classes: Point2D, Circle2D**,** Rect2D, Segment2D, Triangle2D**, Polygon2D**. Note: the polygon is a relatively complex class - make sure to go over the guidance of area, and contains. With respect to the **area** and the **contains** methods - one may assume that the polygon is **simple** (no self intersection).
5. Implement and update the related needed classes, in particular: Ex4, GUIShape and ShapeCollection (you can add additional classes if needed).
6. Implement detailed JUnit classes (this time there are no skeleton classes given to you - make sure you implement a complete testing suite in classes: Point2DTest, Rect2DTest, Segment2DTest, Triangle2DTest, Polygon2DTest, GUIShapeTest, ShapeCollectionTest, Ex4Test
7. **Add a detailed documentation (in English) to the Ex4.java** file with the related description for each function.
8. Make sure to submit ALL the needed classes + an “executable” jar fine named Ex4.jar that can be run by double clicking.

**Notes**:

1. Work in pairs (or alone)! - you can talk about this assignment with anyone in class - but when writing your solution DIY!. Please go over this [document](https://www.ariel.ac.il/wp/cs/wp-content/uploads/sites/88/2020/08/Guidelines-for-Academic-Integrity.pdf) which covers the School’s honesty policy. Make sure to read the remarks regarding chatGPT at the end of this document.
2. **Make sure you write your IDs (ID1 & ID2) in the files Ex4.java**
3. The implementation of the function should be as efficient and elegant as possible.
4. **Keep in mind: we have “planted” few minor “bugs” in the suggested jar file ⇒ in order to force you to use proper testing! (and not just comparing with our solution).**
5. Your solution should be submitted to Moodle according to the instructions - as presented to you in the TA sessions.

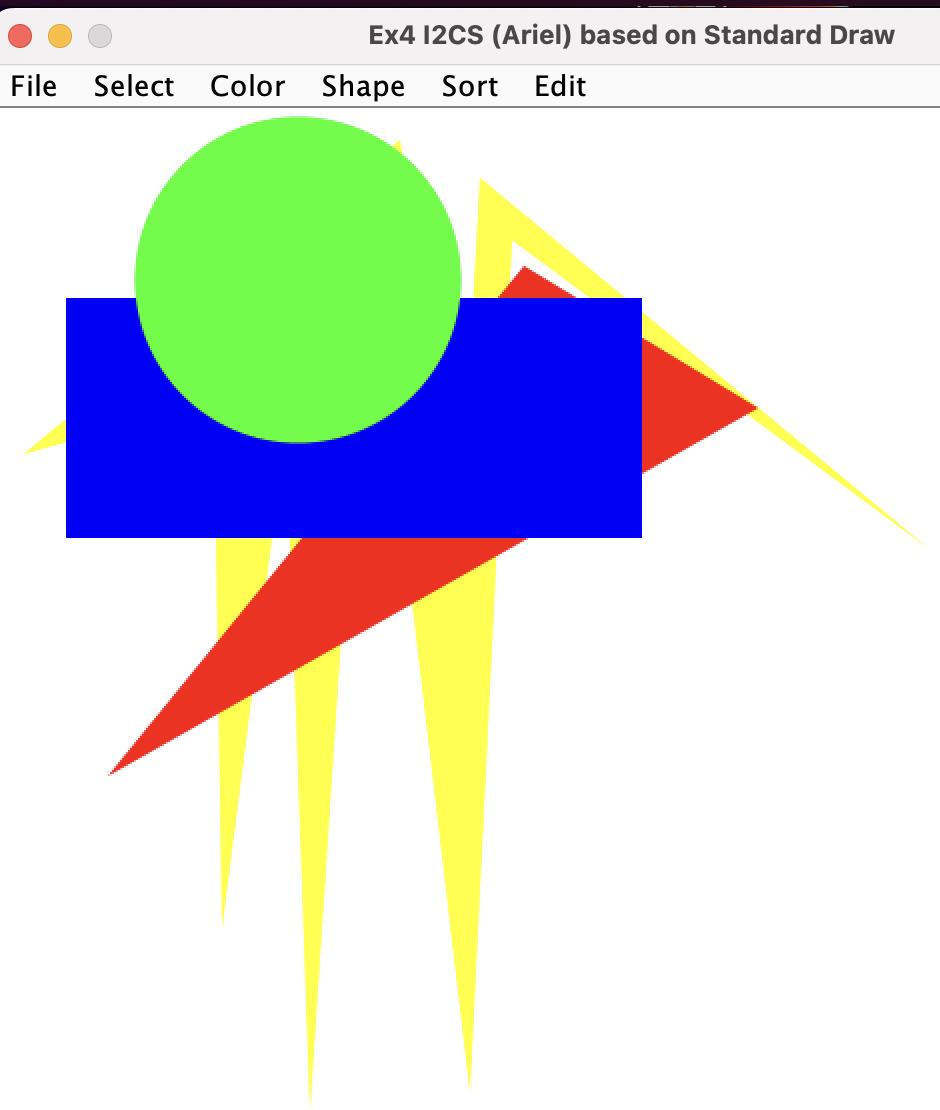
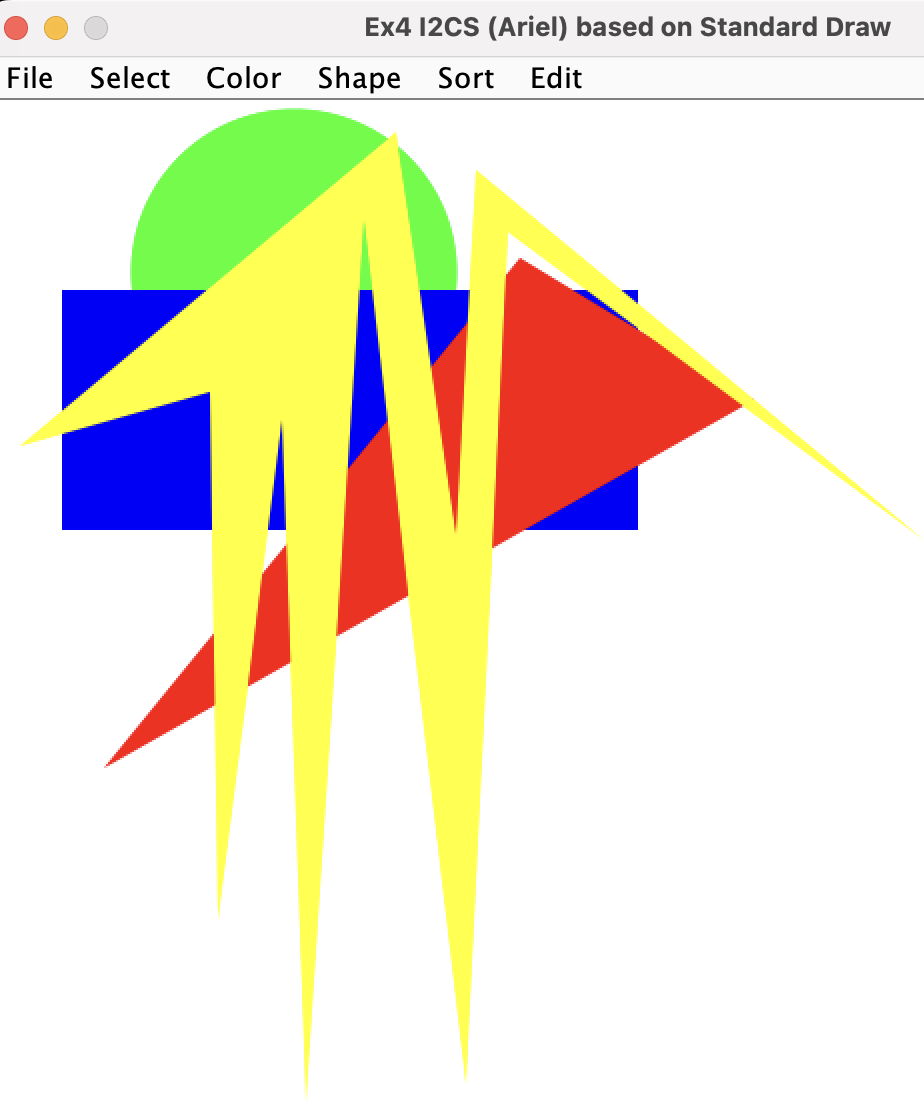


Figure 1: The “a0” file, as opened in the runnable jar. Left ordered by “perimeter” (yellow top - largest). Right: sorted by “antiPerimeter” (green top - smallest). Make sure you play with all tha options - including the sorting, scaling, rotation,save & load.

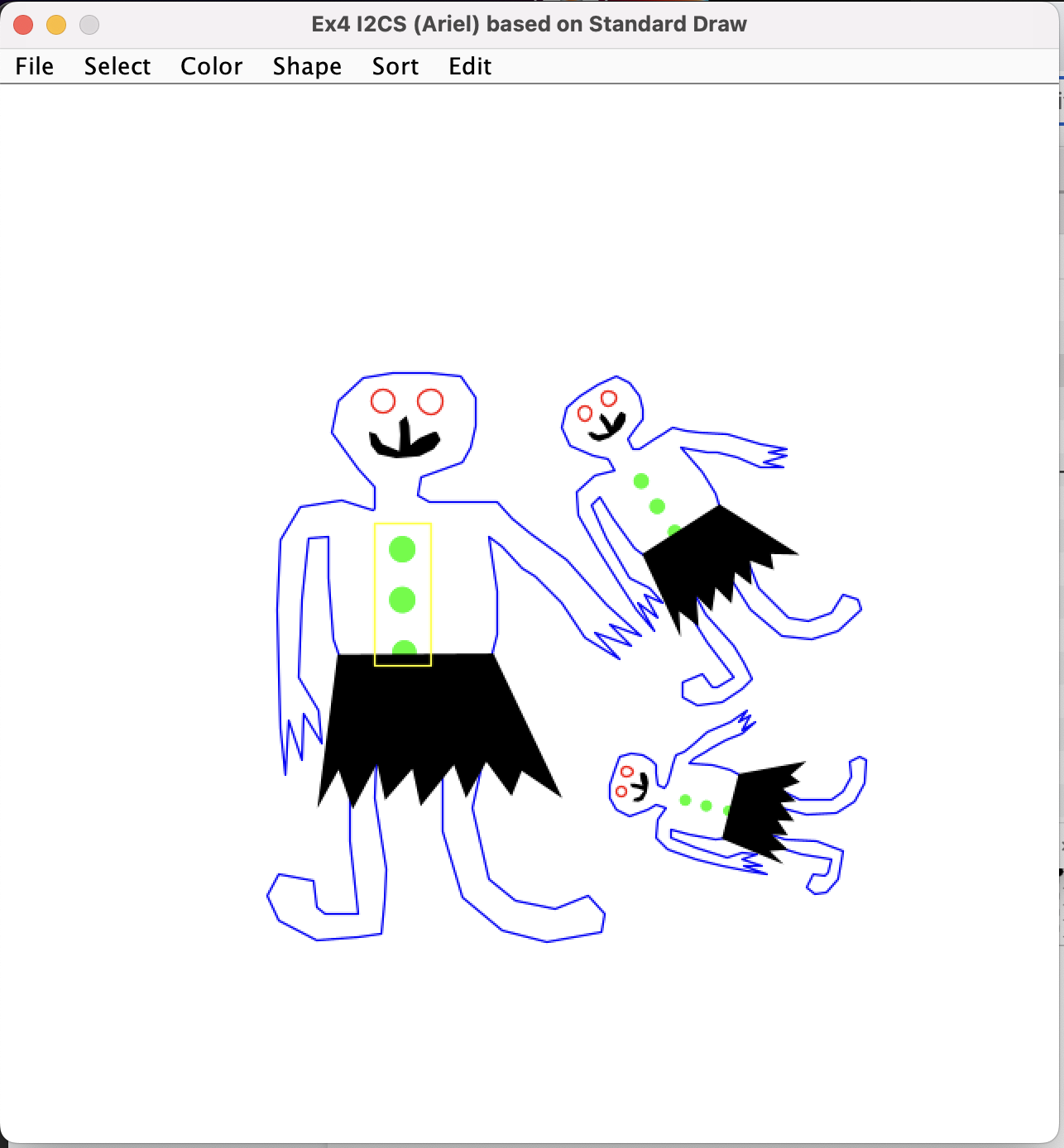


Figure 2: file “a2” - rotation and scaling.

Links:

* [Ex4 101](https://ariel-ac-il.zoom.us/rec/share/NomzfbZsE0i56zaOWiG_nIKbwdk4oFv7ZSSRzRxjU3jqkkQ82jkXeM-GrqAmEUE.uyxUd1Q7maXezIzE): how to start (a zoom talk regarding Ex4)
* [Ex4 201](https://ariel-ac-il.zoom.us/rec/share/oI0myPil2QusFvTDEX9TYQfKm5rfiwSQ5nV48oH2s7AePwc-cIULXiSASYuCiQdA.fpgZqDZBgip4eVZx?startTime=1672243261000): Q & A (Comparator, Polygon, Rotate…)
* Polygon: <https://en.wikipedia.org/wiki/Polygon>
* Testing if a polygon contains a point : <https://en.wikipedia.org/wiki/Point_in_polygon>

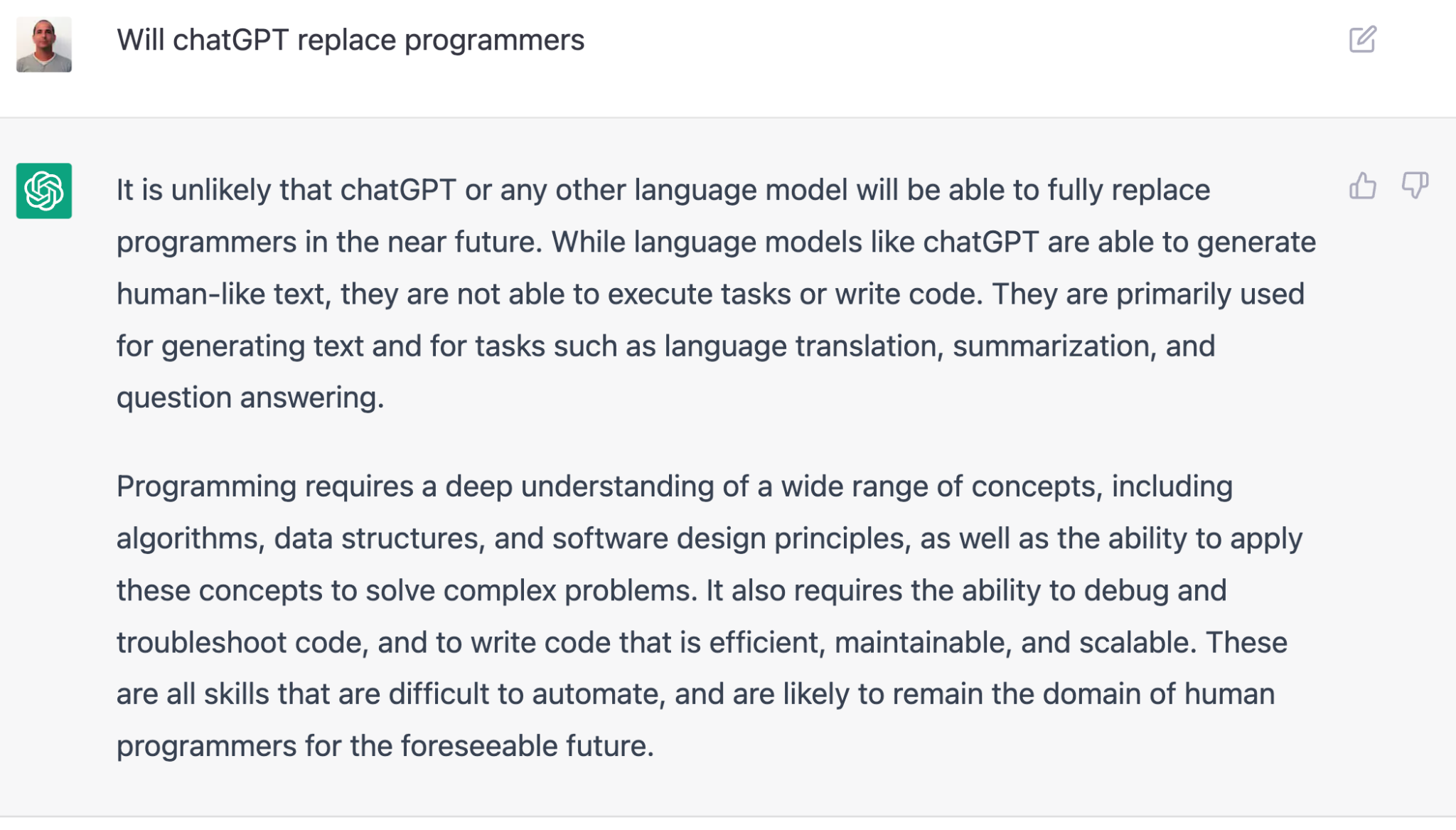
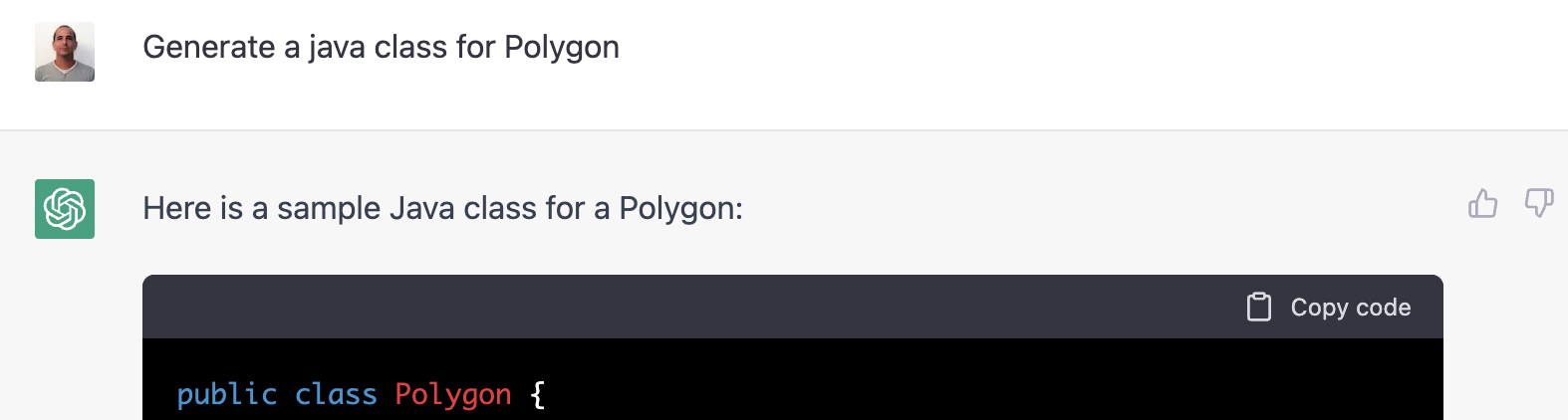
****

Figure 3: chatGPT is a source like any other - make sure to report on any source you are using! → Write your own code!

**Q & A**

1. Comparator:

if(\_flag == Ex4\_Const.Sort\_By\_toString) {

ans = o1.toString().compareTo(o2.toString());

}

if(\_flag == Ex4\_Const.Sort\_By\_Area) {

double d1 = o1.area();

double d2 = o2.area();

if(d1<d2) {ans = -1;}

if(d2<d1) {ans = 1;}

}

<https://docs.oracle.com/en/java/javase/12/docs/api/java.base/java/util/Arrays.html#sort(T%5B%5D,java.util.Comparator)>

<https://docs.oracle.com/javase/7/docs/api/java/util/Collections.html#sort(java.util.List,%20java.util.Comparator)>

1. Rotate & Scale (center Point)

<https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html#atan2-double-double->

1. Polygon:
   1. Point inside, given a point p and a polygon Po: is Po containing p:

define a outer point Pmin1(Xmin-1, Ymin-1), Pmin2(Xmin-1, p.y);

* 1. area: <https://www.baeldung.com/cs/2d-polygon-area>