

## Part B

1. העמסה

Triangle1 and Triangle2:

קיימים שני בנאים: אחד דיפולטיבי ואחד שמקבל ערכים.  
קיימות 2 מטודות contains(Point p), contains(Triangle Tri)  
מחלקת AbsTriangle אבא לTriangle1, Triangle2  
קיימות 2 מטודות contains(Point p), contains(Triangle tri1)

2. פולימורפיזם

Triangle1, Triangle2 is an AbsTriangle and a Triangle

3. עיקרון הכימוס

חסימת הגישה לשדות או מטודות בתוך המחלקה ע"י הכרזה private וכך המשתמש לא יוכל לגשת לשדות או מטודות אלו.  
במחלקות Triangle1, Triangle2:  
כל השדות הם private.  
במחלקת AbsTriangle:  
יש מטודה private שנקראת y().

4. לו המטודה getCenter() הייתה מחזירה מצביע לנקודה ולא מצביע לאובייקט חדש אז היו יכולים לשנות את הנקודה ע"י מטודות של המחלקה Point והשינויים היו מתבצעים על הנקודה centerPoint במקביל כי יש להם את אותה כתובת בזיכרון, ולא דרך setter and getter שהם הבקרה שלנו.

```
1
2 public class mainHW2 {
3
4     public static void main(String[] args) {
5         Triangle tri = new Triangle1();
6         Point centerPoint = tri.getCenter();
7         centerPoint.setX(6);
8         //this would change the centerPoint x value in the triangle tri
9     }
10
11 }
12
```

5. א.

מה שבחרנו לא לעשות במשותף, המטודה isEqual() שיותר נוח לממש אותה בהתאם לכל שדה שיש לכל מחלקה ובכך לעשות את ההשוואה, בנוסף היא רק שורה אחת. יכולנו להשוות דברים משותפים ולשים אותה במימוש משותף AbsTriangle.

ב.

עשינו מחלקת אבא אבסטרקטית שמממשת את Triangle והיא מורישה ל Triangle1, Triangle2. המימושים המשותפים נמצאים ב AbsTriangle והמימושים שלא זהים נמצאים ב Triangle1, Triangle2.

ג.

ישנם כמה משתנים שרושמים אותם פעמיים, אם המטודה הייתה בתוך Triangle1 אז לא היינו צריכות לשמור את המשתנים ולייבא אותם כי הם היו כבר השדות שלו. ואותה הדבר בTriangle2