**הנחות עבודה:**

1. הפרמיטיב היחיד האפשרי הוא מסוג אינטג'ר.

לדוגמא – x=3 חוקי, x='a' לא חוקי

1. מטפלים בפניה לאטריוביטים:

x.a.a.a // חוקי

1. הקונבנציה היא שמחלקה מתחילה באות גדולה, פונקציה מתחילה באות קטנה
2. אין side effects – פונקציה מקבלת ערכים ומחזירה ערכים
3. רצים על הקוד כפי שהוא, הקוד אינו מכיל הגדרות כלשהן (פונקציות, מחלקות וכו') ואינו מכיל import והכל מוגדר כהלכה.

דוגמא:

x=Sheker() // מניחים שSheker היא מחלקה שמוגדרת היטב

If x.n > 0: // מניחים שלx יש attribute n והפעולה גדול מ חוקית עליו

y = 5

Else:

y=6

1. מטפלים בקריאה לפונקציה (רגילה או מחלקה) ע"י הנחה שחוזר ערך שאנחנו יודעים לטפל בו. לדוגמא:

result = x.func() // חוקי

result.bling() // חוקי, מניחים שיש באמת פונקציה כנדרש

result.bling + result // חוקי, מניחים שזה עובד

print result // לא חוקי

1. קונסטרקטור/פונקציות יכולות לקבל פרמטרים. הפרמטרים מועברים כפי שהם ולא אטריביוט בתוכם. לדוגמא:

x = Sheker(a, b, c) // חוקי

x = Sheker(a.n) // לא חוקי

1. הפעולה האריתמטית היחידה שמטפלים בה היא פעולת חיבור
2. כל הפעולות הן בינאריות
3. המשתנה לא מוגדר ע"י עצמו:

x = x + 10 // לא חוקי

1. ההשמה היא תמיד למשתנה יחיד:

x, y = sheker() // לא חוקי

1. אין מחזור שמות משתנה
2. מטפלים בif, if else, while, לא מטפלים בדברים מתקדמים כמו list comprehension ושטויות כאלה
3. אם נספיק נוסיף גם תמיכה בException ו-list (נשמע שlist יותר מעניין?)
4. הקלט לתוכנית הוא קובץ פיית'ון שממנו נקראת התוכנית, המשתנה שעליו אנחנו רוצים לעשות את ההטלה ומיקום לפלט
5. במיקום של הפלט יהיו 2 קבצים:
   1. גרף שמתאר את התלויות כמו שראינו בכיתה
   2. התוכנית המוטלת
6. ממשים את האלגוריתם של Chaotic Iteration.

**מהלך העבודה:**

1. Flow בסיסי
2. הוספת if
3. הוספת while
4. הוספת ירושה