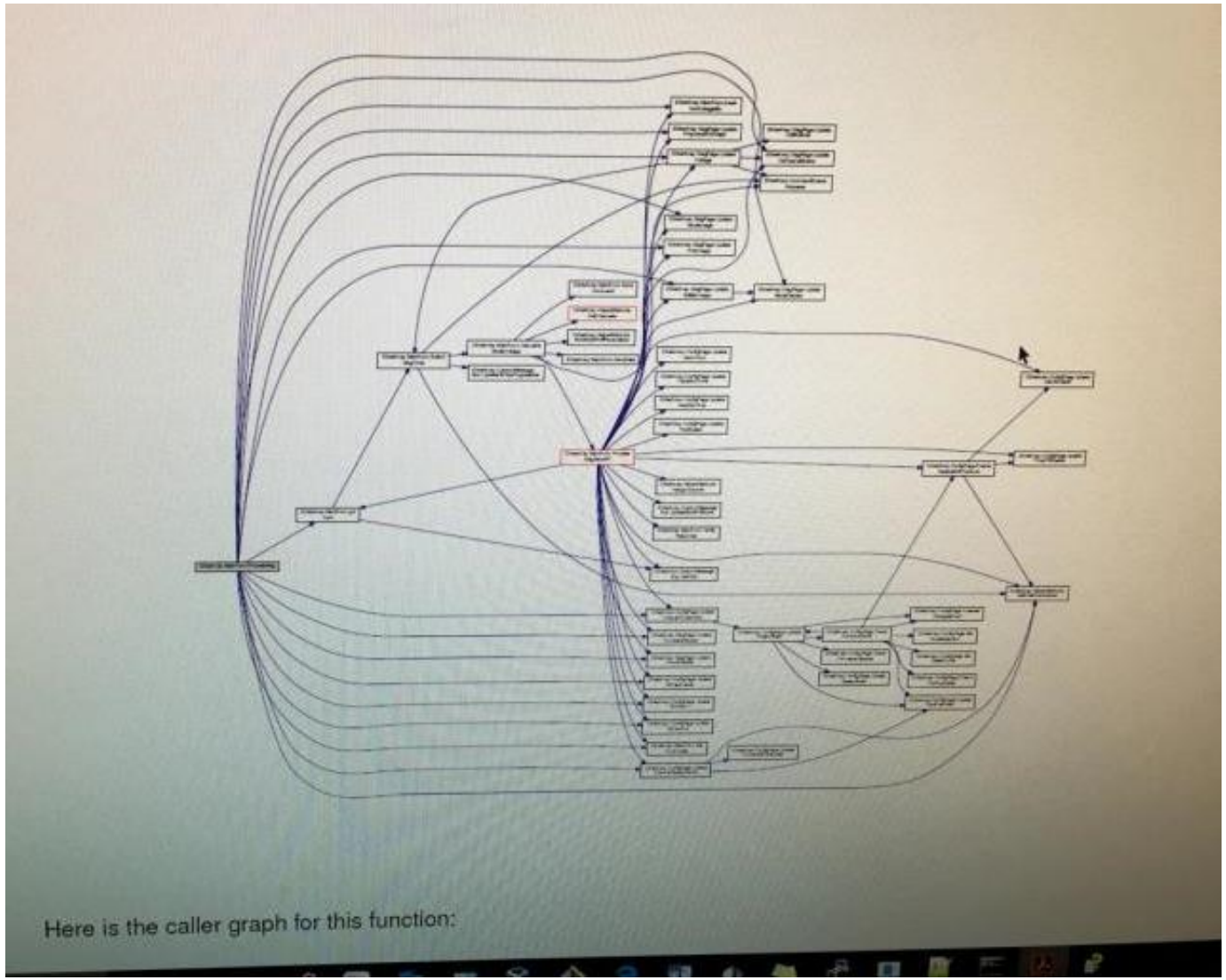
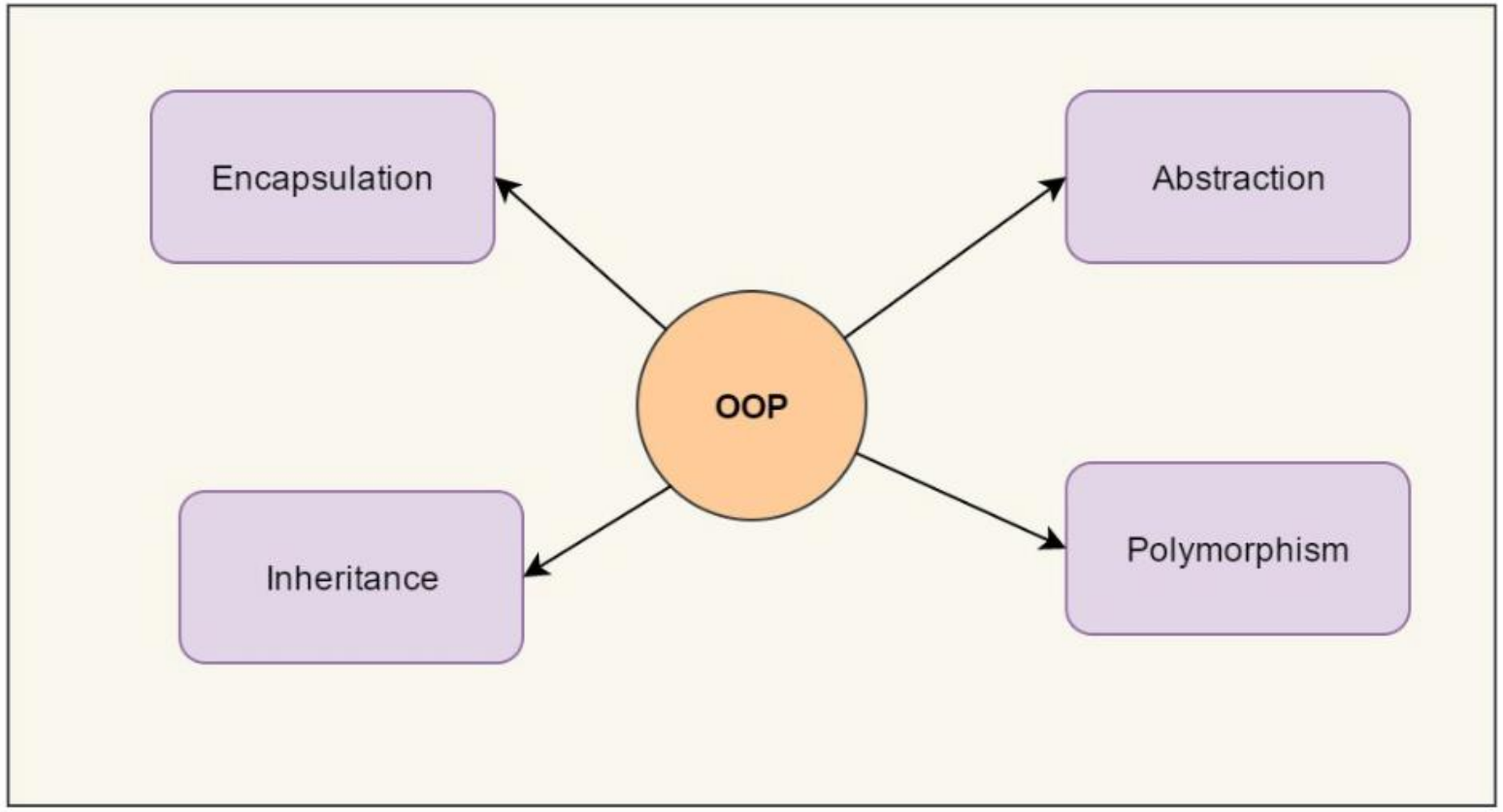
תכנות מונחה עצמים בפייתון

הבעיה בתכנות פרוצדורלי



העקרונות



Four Pillars of Object Oriented Programming

אנקפסולציה - כימוס

- אנחנו לא רוצים לדעת איך דברים נעשים אם אין לנו צורך בזה
- אנחנו רוצים רק ממשק שימושי עם האובייקטים
- לא אכפת לנהג איך האוטו זז

אבסטרקטציה - הפשטה

- אנחנו חושבים על מושגים בהכללה ולא מנסים לבנות חליפה תפורה
- פטיש, צבת מתאים יותר למימוש מחלקה
- אבל המחלקה תהיה כלי עבודה
- יותר מופשט מזה "מוצרים" רק עם מחיר וכן הלאה

הורשה

- קשור להפשטה, כמו מוצרים > כלי עבודה > פטיש
- as a vs has a

פולימורפיזם רב צורתיות

- צורות רבות בממשק אחד
- למשל ראינו את זה ב (x, *args, **kwargs)

מחלקות בפייתון

- כך נגדיר

```
class my_class:
        pass
```

- נגדיר לו initializer

```
class my_class:
        def __init__(self, x, y):
                self.my_x = x
                self.my_y = y
```

דוגמה

```
class yom_tov:
        meals = True
        def __init__(self, name, length):
                self.name = name
                self.length = length
        def prepare(self):
                if meals:
                        print(str(length*2) + “ meals”)
                else:
                        print(“nothing”)
```

הורשה

```
class yom_tov_deorayse(yom_tov):
        pass
```

דריסה

```
class yom_tov_deorayse(yom_tov):
        def __init__(self,name, season, length, cook):
                yom_tov.__init__(self,name, season, length)
                self.cook = cook

        def has_hallel():
                if cook & length != 2:
                        return True
                return False
```

תרגיל על סינגלטון...