**תרגול – Inheritance, Override, Polymorphism**

בתרגיל זה נכתוב תוכנית לניהול כח אדם בחברת תוכנה.

כתבו את המחלקות הבאות והריצו את ה-main לדוגמא שמופיע בסוף התרגיל.  
תוודאו שאתם מקבלים תוצאות דומות בפורמט של הפלט לדוגמא .  
תריצו מספר פעמים כדי לוודא שטיפלתם בכל האפשרויות.

בהצלחה רבה.

1. המחלקה **Employee**

כתבו מחלקה חדשה Employee ע"פ הדרישות הבאות:

1. Attributes – מאפיינים
   * id => automat id start by 1 with prefix emp, e.g. emp\_1
   * firstname
   * lastname
   * email => format: firstname.lastname@email.com
   * salary
2. Methods – מתודות
   * \_\_init\_\_(firstname, lastname, salary)
   * \_\_str\_\_()
   * \_\_eq\_\_(other): boolean  
     compare 2 employees by id
   * apply\_raise() => raising employee’s salary by 4 percent

**שימו לב**

* התכונה id מקבלת ערך אוטומטי החל מ-1 עבור כל מופע של עובד שנוצר, האורח הראשון יהיה 1 ואחריו 2 וכן הלאה.
* השיטה apply\_raise מעלה את שכר העובד/ת ב-4%.

1. המחלקה **Developer**

כתבו מחלקה חדשה Developer אשר יורשת מהמחלקה Employee ע"פ הדרישות הבאות:

1. Attributes – מאפיינים
   * prog\_langs : []
2. Methods – מתודות
   * \_\_init\_\_( firstname, lastname, salary, prog\_langs)
   * \_\_str\_\_()

**שימו לב**

* התכונה prog\_langs עבור רשימת שפות התכנות שהעובד המתכנת שולט בהן, ערך ברירת המחדל של תכונה זאת הינו רשימה ריקה.
* השיטה apply\_raise עבור עובד מתכנת מעלה את שכר העובד/ת ב-10%.
* השיטה str תוסיף את הטקסט :’is a developer’

1. המחלקה **Manager**

כתבו מחלקה חדשה Manager אשר יורשת מהמחלקה Employee ע"פ הדרישות הבאות:

1. Attributes – מאפיינים
   * emploees : []
2. Methods – מתודות
   * \_\_init\_\_( firstname, lastname, salary, emploees)
   * \_\_str\_\_()
   * add\_employee(employee)
   * remove\_employee(employee)
   * print\_employees()
   * print\_employees\_salary()
   * apply\_all\_raise() => raising all employee’s salary

**שימו לב**

* התכונה emploees עבור רשימת העובדים, ערך ברירת המחדל של תכונה זאת הינו רשימה ריקה.
* השיטה str תוסיף את הטקסט :’is a manager’
* השיטה apply\_raise עבור מנהל מעלה את שכר העובד/ת ב-15%.
* השיטה apply\_all\_raise תעלה את שכר כל העובדים של המנהל.

main לדוגמא:

def main():  
 employees\_details = [  
 ['', 'Mor', 'Avner', 10000],  
 ['developer', 'Moshe', 'Cohen', 20000, ['java', 'python', 'c++']],  
 ['developer', 'Kobi', 'Shalom', 22000, ['java', 'nodeJs']],  
 ['', 'Tom', 'Keren', 12000],  
 ['developer', 'Liat', 'Barak', 30000, ['java', 'python', 'javascript', 'dart']]  
 ]  
 employees = []  
 for item in employees\_details:  
 if item[0] == 'developer':  
 employees.append(Developer(item[1], item[2], item[3], item[4]))  
 else:  
 employees.append(Employee(item[1], item[2], item[3]))  
  
 manager = Manager('Galit', 'Mor', 35000, employees)  
 print(manager)  
 manager.print\_employees\_salary()  
 manager.apply\_all\_raise()  
 print('After apply salary')  
 manager.print\_employees\_salary()  
 manager.add\_employee(Developer('Yael', 'Marom', 20000, ['java', 'angular']))  
 print('After adding new employee')  
 manager.print\_employees\_salary()

פלט לדוגמא:

Galit Mor is a manager

Mor Avner 10000

Moshe Cohen is a developer 20000

Kobi Shalom is a developer 22000

Tom Keren 12000

Liat Barak is a developer 30000

After apply salary

Mor Avner 10400

Moshe Cohen is a developer 22000

Kobi Shalom is a developer 24200

Tom Keren 12480

Liat Barak is a developer 33000

After adding new employee

Mor Avner 10400

Moshe Cohen is a developer 22000

Kobi Shalom is a developer 24200

Tom Keren 12480

Liat Barak is a developer 33000

Yael Marom is a developer 20000