Albert Nguyen
CS131-47853 MW 06:00 PM – 08:30 PM
Programming Assignment 5

Programming Assignment 5 – Chapter 10

Instructions

a.

Make a class Employee with a name and salary. Make a class Manager inherit from Employee. Add an instance variable, named _department, that stores a string. Supply a method __repr__ that prints the manager's name, department, and salary. Make a class Executive inherit from Manager. Supply appropriate __repr__ methods for all classes. Supply a test program that tests these classes and methods.

Sample output:
John Smith has a salary of 45000.00
Jane Doe has a salary of 60000.00 and manages the Widgets department
Weird Guy has a salary of 90000.00 and is the executive for the Thingies Department

Code

employees.py

employees.py | employees_demo.py
Executive | __repr__

```
1  class Employee:
2      def __init__(self, name, salary):
3          self._name = name
4          self._salary = salary
5
6      def __repr__(self):
7          return "%s has a salary of %.2f" % (self._name, self._salary)
8
9
10 class Manager(Employee):
11     def __init__(self, name, salary, department):
12         super().__init__(name, salary)
13         self._department = department
14
15     def __repr__(self):
16         return "%s has a salary of %.2f and manages the %s department"\
17                % (self._name, self._salary, self._department)
18
19
20 class Executive(Manager):
21     def __init__(self, name, salary, department):
22         super().__init__(name, salary, department)
23
24     def __repr__(self):
25         return "%s has a salary of %.2f and is executive for the %s department"\
26             % (self._name, self._salary, self._department)
27
```

Albert Nguyen
CS131-47853 MW 06:00 PM – 08:30 PM
Programming Assignment 4

employeesDemo.py (import PA_Car.py Car class)

employees.py employees_demo.py

(bottom)

```
1  from employees import *
2
3
4  def main():
5      employeeDemo = Employee("John Smith", 45000)
6      managerDemo = Manager("Jane Doe", 60000, "Widgets")
7      executiveDemo = Executive("Weird Guy", 90000, "Thingies")
8
9      print(employeeDemo)
10     print(managerDemo)
11     print(executiveDemo)
12
13 main()
14
```

Output

Python Shell: Wing

Python Shell

Commands execute without debug. Use arrow keys for history.

```
   Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
   Type "help", "copyright", "credits" or "license" for more information.
>>> [evaluate employees_demo.py]
   John Smith has a salary of 45000.00
   Jane Doe has a salary of 60000.00 and manages the Widgets department
   Weird Guy has a salary of 90000.00 and is executive for the Thingies department
>>>
```

Albert Nguyen
CS131-47853 MW 06:00 PM – 08:30 PM
Programming Assignment 5

Lab 9 – Written Code

employees.py

```
class Employee:
    def __init__(self, name, salary):
        self._name = name
        self._salary = salary

    def __repr__(self):
        return "%s has a salary of %.2f" % (self._name, self._salary)


class Manager(Employee):
    def __init__(self, name, salary, department):
        super().__init__(name, salary)
        self._department = department

    def __repr__(self):
        return "%s has a salary of %.2f and manages the %s department"\
               % (self._name, self._salary, self._department)


class Executive(Manager):
    def __init__(self, name, salary, department):
        super().__init__(name, salary, department)

    def __repr__(self):
        return "%s has a salary of %.2f and is executive for the %s department"\
            % (self._name, self._salary, self._department)
```

employees_demo.py

```
from employees import *


def main():
    employeeDemo = Employee("John Smith", 45000)
    managerDemo = Manager("Jane Doe", 60000, "Widgets")
    executiveDemo = Executive("Weird Guy", 90000, "Thingies")

    print(employeeDemo)
    print(managerDemo)
    print(executiveDemo)

main()
```