

Object Oriented Programming LAB – BSDSF22

(Morning and Afternoon)

Lab 07 – 20-10-2023

A company stores *employee number*, *names*, *cnic numbers*, and contact details (*phone* and *address*) of all of its **employees** and pays them on a weekly basis. The employees are of three types: **Salaried** employees are paid a fixed weekly salary regardless of the number of hours worked, **hourly** employees are paid by the hour and receive overtime pay for all hours worked in excess of 40 hours, and **commission** employees are paid a percentage of their sales.

Specific data members for salaried employee are *weekly salary*, hourly employee are *hours worked* and *rate per hour*, and commission employee are *sales total* and *percentage*.

Main task: You have to program the company's payroll calculations (a list of all employees with the amount to pay for the week).

Note the following important points:

1. At the beginning, you must focus on inheritance/polymorphism here, ignore composition and aggregation for this task.
2. You must implement all required constructors and dunder (magic methods).
3. To save time, make data members public to avoid writing of getters/setters and input/output functions.
4. To save provide implementation any two of derived classes of **Employee**.
5. You may have to create a class **Payroll** to compose/aggregate **Employees**.
6. You MUST have to write driver code (or main function) for sufficient hard coded employee's data.

This lab has resemblance with shapes case study in the following ways:

1. **Employee** class corresponds to shape class, the base of derived classes **Salaried**, **Rated**, or **Commissioned**.
2. Each of the class **Salaried**, **Rated**, or **Commissioned** has a function **calculatePay** to compute weekly payment, like *area* function in **Rectangle** and **Circle** classes.
3. **Payroll** class is like the Canvas class with various arrays and their filled counts.

Bonus task: Update your code if all the employees belongs to any of the following departments: sales located in Lahore, accounting located in Kasur, manufacturing located in Faisalabad. [it is an aggregation]

***** The End *****

Just think the definition of the class named **ROBOT** with following functionality. The robot is associated with rectangular floor conceptually divided in rows and columns, with some have obstacles.

```
getRobotName() : str
setRobotName(n: str) : None
getRowNumber() : int
setRowNumber(v: int) : None
getColumnNumber() : int
setColumnNumber(v: int) : None
getDirection() : str
voidsetDirection(d: str) : None
```

```
setRobot(nm: str, cx: int, cy:
        int, dr: str) : None
canStep(front: str) : bool
takeStep() : None
turnLeft() : None
turnRight() : None
turnBackward() : None
show() : None    # may be str
```

Bonus task: Draw UML Class for ROBOT and Board classes.