**Lab Report – 02**

1. **Write a Python class Employee with attributes like emp\_id, emp\_name,**

**emp\_salary, and emp\_department and methods like calculate\_emp\_salary,**

**emp\_assign\_department, and print\_employee\_details.**

**Sample Employee Data:**

**"ADAMS", "E7876", 50000, "ACCOUNTING"**

**"JONES", "E7499", 45000, "RESEARCH"**

**"MARTIN", "E7900", 50000, "SALES"**

**"SMITH", "E7698", 55000, "OPERATIONS"**

* **Use 'assign\_department' method to change the department of an employee.**
* **Use 'print\_employee\_details' method to print the details of an employee.**
* **Use 'calculate\_emp\_salary' method takes two arguments: salary and**

**hours\_worked, which is the number of hours worked by the employee. If the**

**number of hours worked is more than 50, the method computes overtime and adds it to the salary. Overtime is calculated as following formula:**

**overtime = hours\_worked – 50**

**Overtime amount = (overtime \* (salary / 50))**

**Code:**

class Employee:

"""Class to represent an employee."""

def \_\_init\_\_(self, emp\_id, emp\_name, emp\_salary, emp\_department):

"""Initialize the employee object."""

self.emp\_id = emp\_id

self.emp\_name = emp\_name

self.emp\_salary = emp\_salary

self.emp\_department = emp\_department

def calculate\_emp\_salary(self, hours\_worked):

"""Calculate the employee's salary, including overtime pay if applicable."""

overtime = max(hours\_worked - 50, 0)

overtime\_pay = overtime \* (self.emp\_salary / 50)

total\_salary = self.emp\_salary + overtime\_pay

return total\_salary

def emp\_assign\_department(self, new\_department):

"""Assign the employee to a new department."""

self.emp\_department = new\_department

def print\_employee\_details(self):

"""Print the employee's details to the console."""

print(f"Employee ID: {self.emp\_id}")

print(f"Employee Name: {self.emp\_name}")

print(f"Employee Salary: {self.emp\_salary}")

print(f"Employee Department: {self.emp\_department}")

1. **Write a Python class BankAccount with attributes like account\_number,**

**balance, date\_of\_opening and customer\_name, and methods like deposit, withdraw, and check\_balance.**

**Code:**

class BANKACCOUNT:

def \_init\_(self,account\_no,balance,date\_of\_opening,customer\_name):

self.account\_no=account\_no

self.balance=balance

self.date\_of\_opening=date\_of\_opening

self.customer\_name=customer\_name

def Account\_detail(self):

print("Customer name :",self.customer\_name)

print("Account no :",self.account\_no)

print("Account opening date :",self.date\_of\_opening)

print("Account Balance :",self.balance)

def Deposit(self,damount):

Total\_balance=self.balance+damount

self.balance=Total\_balance

print("Total balance =",Total\_balance)

def withdraw(self,wamount):

Total\_Balance=self.balance-wamount

self.balance=Total\_Balance

print("Total balance =",Total\_Balance)

def Check\_balance(self):

print("Your Total Balance is :",self.balance)

B=BANKACCOUNT(3730, 50000,"2-09-2023","Maroof Baloch")

B.Account\_detail()

print("What you want to do press 1 for withdrawl 2 for deposit and 3 for checking balance")

a=int(input())

if(a==1):

print("how much amount you want to withdraw ")

b=int(input())

B.withdraw(b)

B.Account\_detail()

elif(a==2):

print("how much you want to deposit ")

am=int(input())

B.Deposit(am)

B.Account\_detail()

elif(a==3):

B.Check\_balance()

B.Account\_detail()

else:

print("Wrong choice please select from 1,2or3")