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}")
```

2. 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")
```