**Section-1**

Python:

Q-1)

filename=input("write the name of the file which should be opened :")

search=input("word to be searched")

occurance=0

with open(filename,'r') as fn:

for line in fn:

k=line.split()

for i in k:

if(i==search):

occurance=occurance+1

print("no of occurence ",occurance)

**Section-2:**

Unix:

read -p "enter name of the file: " fn

if [ -e$fn ]; then

echo "File Found"

count=`echo |wc -l $fn`

echo "Number of Lines=$count"

unique\_lines=`echo| sort $fn|uniq`

echo "after deleting duplicates, remaining lines= $unique\_lines"

else

echo "File Not Found"

fi

**Section-3:**

Oracle:

Q1)

1 🡪 SELECT (E.FIRST\_NAME||' '||E.LAST\_NAME) AS NAME,(E1.FIRST\_NAME||' '||E1.LAST\_NAME) AS MANAGER\_NAME,B.NAME,D.NAME FROM EMPLOYEE E INNER JOIN EMPLOYEE E1 ON E.SUPERIOR\_EMP\_ID=E1.EMP\_ID INNER JOIN BRANCH B ON E.ASSIGNED\_BRANCH\_ID=B.BRANCH\_ID AND E.ASSIGNED\_BRANCH\_ID IN(1,2) INNER JOIN DEPARTMENT D ON E.DEPT\_ID=D.DEPT\_ID;

2🡪 select \* from Acc\_transaction where to\_char(Txn\_Date,'yyyy')=2002;

3🡪 select \* from Customer,ACCOUNT where Customer.Cust\_id=ACCOUNT.Cust\_id and ACCOUNT.Open\_date=ACCOUNT.Last\_activity\_date;

4🡪 SELECT (E.FIRST\_NAME||' '||E.LAST\_NAME) AS EMP\_NAME,B.NAME,A.CUST\_ID,A.ACCOUNT\_ID FROM ACCOUNT A INNER JOIN ACCOUNT A1 ON A.AVAIL\_BALANCE=A1.PENDING\_BALANCE INNER JOIN EMPLOYEE E ON A.OPEN\_EMP\_ID=E.EMP\_ID INNER JOIN BRANCH B ON E.ASSIGNED\_BRANCH\_ID=B.BRANCH\_ID;

Q-2)

declare

procedure p is cursor c is select \* from ACCOUNT;

v\_acc ACCOUNT%rowtype;

begin

if not c %isopen then

open c;

end if;

loop

fetch c into v\_acc;

dbms\_output.put\_line(v\_acc.Avail\_balance || ' ' || v\_acc.Pending\_balance || ' ' || v\_acc.Avail\_balance - v\_acc.Pending\_balance);

exit when c %notfound;

end loop;

close c;

end;

begin

p();

end;

/

Q-3)

declare

procedure p is cursor c(no int) is select \* from employee where Dept\_id in (1,2);

v\_name Employee%rowtype;

begin

if not c %isopen then

open c(2456);

end if;

loop

fetch c into v\_name;

dbms\_output.put\_line(v\_name.Emp\_id);

exit when c %notfound;

end loop;

close c;

end;

begin

p();

end;

/

Q-4)

create or replace procedure proc(p\_id in Acc\_transaction.Account\_id%type) is

cursor c is select \* FROM Account\_transaction where Account\_id = p\_id;

v c%rowtype;

newamt Acc\_transaction.Amount%type;

BEGIN

IF NOT C %ISOPEN THEN

OPEN C;

END IF;

fetch c into v;

UPDATE Acc\_transaction SET Amount = (1.1 \* Amount) WHERE Account\_id = p\_id;

dbms\_output.put\_line (p\_id ||' '||v.amount||' '||' '||v.funds\_Avail\_date||' ' ||v.txn\_date||' '||' '||v. execution\_branch\_id||' '|| v.txn\_id);

CLOSE C;

proc();

END;

/

**Section-4:**

Java:

Q-1)

package Moduleexam;

public class Employee {

int empId;

String empName;

double sal;

double pFund;

double hAllowance;

public Employee(int empId, String empName, double sal) {

super();

this.empId = empId;

this.empName = empName;

setSal(sal);

}

public int getEmpId() {

return empId;

}

public void setEmpId(int empId) {

this.empId = empId;

}

public String getEmpName() {

return empName;

}

public void setEmpName(String empName) {

this.empName = empName;

}

public double getSal() {

return sal;

}

public void setSal(double sal) {

this.sal = sal;

setpFund();

setHAllowance();

}

public double getpFund() {

return pFund;

}

private void setpFund() {

this.pFund = this.sal \* 0.1;

}

public double gethAllowance() {

return hAllowance;

}

private void setHAllowance() {

this.hAllowance = this.sal \* 0.2;

}

@Override

public String toString() {

return "Employee [empId=" + empId + ", empName=" + empName + ", sal=" + sal + ", pFund=" + pFund + ", hAllowance="

+ hAllowance + "]";

}

}

//testcase

package Moduleexam;

public class Employeetest {

public static void main(String args[]) {

Employee emp = new Employee(46008992, "Yashwanth" , 1000000);

double netSalary = emp.getSal() - emp.getpFund() - emp.gethAllowance();

System.out.println("Net Salary = " + netSalary);

System.out.println(emp);

}

}