**DAY 1 Assignments(11-01-2021)**

1. Select employee\_id, concat(first\_name,last\_name) as full\_name, salary as basic\_salary,(Salary\*0.1) as HRA, (salary\*0.05) as DA, (salary\*0.07) as PF, (salary+ (Salary\*0.1)+ (salary\*0.05)-( salary\*0.07)) as gross\_salary from employees;
2. Select employee\_name, salary, hire\_date from employees where hire\_date between ‘1990-01-01’ to ‘1995-05-05’.
3. **Using Where clause:**

select emp.first\_name, dept.department\_name, loc.city from employees emp, departments dept, locations loc where emp.department\_id=dept.department\_id and dept.location\_id=loc.location\_id;

**Using on clause:**

select emp.first\_name, dept.department\_name, loc.city from (employees emp join departments dept on emp.department\_id=dept.department\_id) join locations loc on dept.location\_id=loc.location\_id;

1. Display details of jobs where the minimum salary is greater than 10000.

select first\_name, hire\_date from employees where hire\_date between 2002 and 2005;

1. Display first name and join date of the employees who is either IT Programmer or Sales Man.

select first\_name, hire\_date from employees where job\_id in('IT\_PROG','SA\_MAN');

1. Display first name, salary, commission pct, and hire date for employees with salary less than 10000.

select first\_name, hire\_date,salary,commission\_pct from employees where salary<10000;

1. Display employees where the first name or last name starts with S.

select first\_name,last\_name from employees where first\_name like's%' or last\_name like 's%';

1. Display details of the employees where commission percentage is null and salary in the range 5000 to 10000 and department is 30.

select \* from employees where commission\_pct is null and salary between 5000 and 10000 and department\_id=30;

1. Display job title, employee ID, number of days between ending date and starting date for all jobs in department 30 from job history.

select j1.job\_title, jh.employee\_id, datediff(jh.end\_date, jh.start\_date) from jobs j1, job\_history jh where j1.job\_id= jh.job\_id;

1. Display country name, city, and department name.

select con.country\_name, loc.city, dept.department\_name from countries con, locations loc, departments dept where con.country\_id=loc.country\_id and loc.location\_id=dept.location\_id;

1. Display employee name and country in which he is working.

select concat(first\_name,' ', last\_name) as employee\_name, country\_name from employees join departments using(department\_id) join locations using(location\_id) join countries using(country\_id);

1. Display details of jobs where the minimum salary is greater than 10000.

select job\_id, job\_title from jobs where min\_salary>10000;

1. Display job Title, the difference between minimum and maximum salaries for jobs with max salary in the range 10000 to 20000.

select job\_id,(max\_salary-min\_salary) from jobs where max\_salary between 10000 and 20000;

1. Display details of jobs in the descending order of the title.

select job\_id, job\_title from jobs order by job\_title desc;

1. Display employees who joined in the month of May.

select concat(first\_name,' ',last\_name) as employee\_name from employees where hire\_date like '\_\_\_\_-05-\_\_%';

1. Display first name, salary, and round the salary to thousands.

select first\_name,round(salary,-3) from employees;

1. Display first name and date of first salary of the employees.

select first\_name,last\_day(hire\_date) from employees;

1. Display first name and experience of the employees.

select first\_name, datediff(sysdate(),hire\_date)/365 as experience from employees;

1. Display the length of first name for employees where last name contain character ‘b’ after 3rd position.

select length(first\_name) from employees where last\_name like '\_\_\_b%';

1. Display first name in upper case and email address in lower case for employees where the first name and email address are same irrespective of the case.

select upper(first\_name),lower(email) from employees where first\_name=email;

1. Display the number of days between system date and 1st January 1995.

select datediff(sysdate(),'1995-01-01');

1. Display employee ID and the date on which he ended his previous job.

select employee\_id,max(end\_date) from job\_history where employee\_id in(select employee\_id from job\_history group by 1 having count(employee\_id)>1) group by 1;

1. Display the country ID and number of cities we have in the country.

select country\_id, count(\*) from locations group by country\_id;

1. Display average salary of employees in each department who have commission percentage.

select department\_id,avg(salary) from employees where commission\_pct is not null group by department\_id;

1. Display job ID, number of employees, sum of salary, and difference between highest salary and lowest salary of the employees of the job.

select job\_id, count(\*), sum(salary),max(salary)-min(salary) from employees group by job\_id;

1. Display job ID for jobs with average salary more than 10000.

select job\_id from employees group by job\_id having avg(salary)>10000;

1. Display years in which more than 10 employees joined.

select date\_format(hire\_date,'%y') from employees group by date\_format(hire\_date,'%y') having count(employee\_id)>10;

1. Display departments in which more than five employees have commission percentage.

select department\_name from departments where department\_id=(select department\_id from employees group by department\_id having count(commission\_pct)>5);

1. Display department name and number of employees in the department.

select department\_name, count(\*) from departments inner join employees on employees.department\_id=departments.department\_id group by departments.department\_id, department\_name order by department\_name;

1. Display employee ID for employees who did more than one job in the past.

select employee\_id from employees where employee\_id not in(select employee\_id from job\_history);

**Day 2 Assignments(21/01/2021):**

1. Change salary of employee 115 to 8000 if the existing salary is less than 6000.

update employees set salary=8000 where employee\_id=115 and salary<6000;

1. Insert a new employee into employees with all the required details.

insert into employees(employee\_id,first\_name,last\_name,email,phone\_int,hire\_date,job\_id,salary,department\_id) values(207,'sam','levis','sam','216 235 1234',sysdate(),'sa\_man',15000,50);

1. Change job ID of employee 110 to IT\_PROG if the employee belongs to department 10 and the existing job ID does not start with IT.

update employees set job\_id='IT\_PROG' where employee\_id=110 and department\_id=10 and not job\_id like 'IT%';

1. Insert a row into departments table with manager ID 120 and location ID in any location ID for city Tokyo.

insert into departments values(280,'sports',120,1200);

1. Display job title, employee ID, number of days between ending date and starting date for all jobs in department 30 from job history.

select job\_title,employee\_id,end\_date-start\_date from job\_history join jobs where department\_id=30;

1. Table ---> Customer

custId, firstName,lastName,age,city, mobileNumber, dob

Add the Constraints

custId is Primary Key

firstName not null

age must be greater than 21

mobile must be unique .

create table customer(custid int primary key,firstname varchar(10) not null,age int check(age>21),mobilenumber int unique,dob int);

1. Table ----> Branch

branchId, branchName, city

Add the Constraints

branchId is Primary Key

branchName not null

city not null.

create table branch(branchid int primary key,branchname varchar(10) not null,city int not null);

1. Table ----> Transaction

transactionId, transactionDate, MediumOfTransaction, TransactionAmount

Add the Constraints

transactionId is primary key.

create table transaction(tranid int primary key,trandate int,mediumoftran varchar(10),tranamount int);

1. Display details of departments managed by ‘John’.

select \* from departments where manager\_id in(select employee\_id from employees where first\_name='john');

1. Display employees who did not do any job in the past.

select \* from employees where employee\_id not in(select employee\_id from job\_history);

1. Display job title and average salary for employees who did a job in the past.

select job\_title,avg(salary) from jobs natural join employees group by job\_title where employee\_id in(select employee\_id from job\_history);

1. Display country name, city, and number of departments where department has more than 5 employees.

select country\_name,city,count(department\_id) from countries join locations using(country\_id) join departments using(location\_id) where department\_id in(select department\_id from employees group by department\_id having count(department\_id)>5) group by country\_name,city;

1. Display details of manager who manages more than 5 employees.

select first\_name from employees where employee\_id in(select manager\_id from employees group by manager\_id having count(\*)>5);

1. Display details of current job for employees who worked as IT Programmers in the past.

select \* from jobs where job\_id in(select job\_id from employees where employee\_id in(select employee\_id from job\_history where job\_id='IT\_PROG'));

1. Display the details of employees drawing the highest salary in the department.

select \* from employees where salary=(select max(salary) from employees);

1. Display third highest salary of all employees

select salary from employees order by salary desc limit 2,1;

**Day 3 Assignment(14/01/2021):**

**Assignment 2**

Take n number records through keyboards as Id,Name,Salary,Desg

(array id,name,salary,Desg).

Salary = salary + hra + da – pf;

Hra is 10% salary

Da is 7 % salary

Pf 5 % salary

If desg is manager desg.equals(“Developer”)

15% bonus

If developer 10% bonus

Else

5 % bonus

Id, name, salary( grossSalary +bonus ) and desg.

**Solution:**

import java.util.Scanner;

class Assignment2

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter number of details you want to store:");

int n=sc.nextInt();

int []id= new int[n];

String []name=new String[n];

float []salary=new float[n];

String []desg=new String[n];

for(int i=0;i<n;i++)

{

System.out.println("Enter id of employee "+(i+1));

id[i]=sc.nextInt();

sc.nextLine();

System.out.println("Enter name of employee "+(i+1));

name[i]=sc.nextLine();

System.out.println("Enter salary of employee "+(i+1));

salary[i]=sc.nextFloat();

sc.nextLine();

System.out.println("Enter designation of employee "+(i+1));

desg[i]=sc.nextLine();

System.out.println("\n");

}

for(int k=0;k<n;k++)

{

float hra,da,pf;

hra=salary[k]\*0.1f;

da=salary[k]\*0.07f;

pf=salary[k]\*0.05f;

if(desg[k].equals("manager"))

{

salary[k]=(salary[k]+hra+da-pf)+(salary[k]\*0.15f);

System.out.println("\n");

System.out.println("Details of Employee:"+(k+1));

System.out.println("id="+id[k]);

System.out.println("Name="+name[k]);

System.out.println("Salary="+salary[k]);

System.out.println("Designation="+desg[k]);

}

else if(desg[k].equals("developer"))

{

salary[k]=(salary[k]+hra+da-pf)+(salary[k]\*0.1f);

System.out.println("\n");

System.out.println("Details of Employee:"+(k+1));

System.out.println("id="+id[k]);

System.out.println("Name="+name[k]);

System.out.println("Salary="+salary[k]);

System.out.println("Designation="+desg[k]);

}

else

{

salary[k]=(salary[k]+hra+da-pf)+(salary[k]\*0.05f);

System.out.println("\n");

System.out.println("Details of Employee:"+(k+1));

System.out.println("id="+id[k]);

System.out.println("Name="+name[k]);

System.out.println("Salary="+salary[k]);

System.out.println("Designation="+desg[k]);

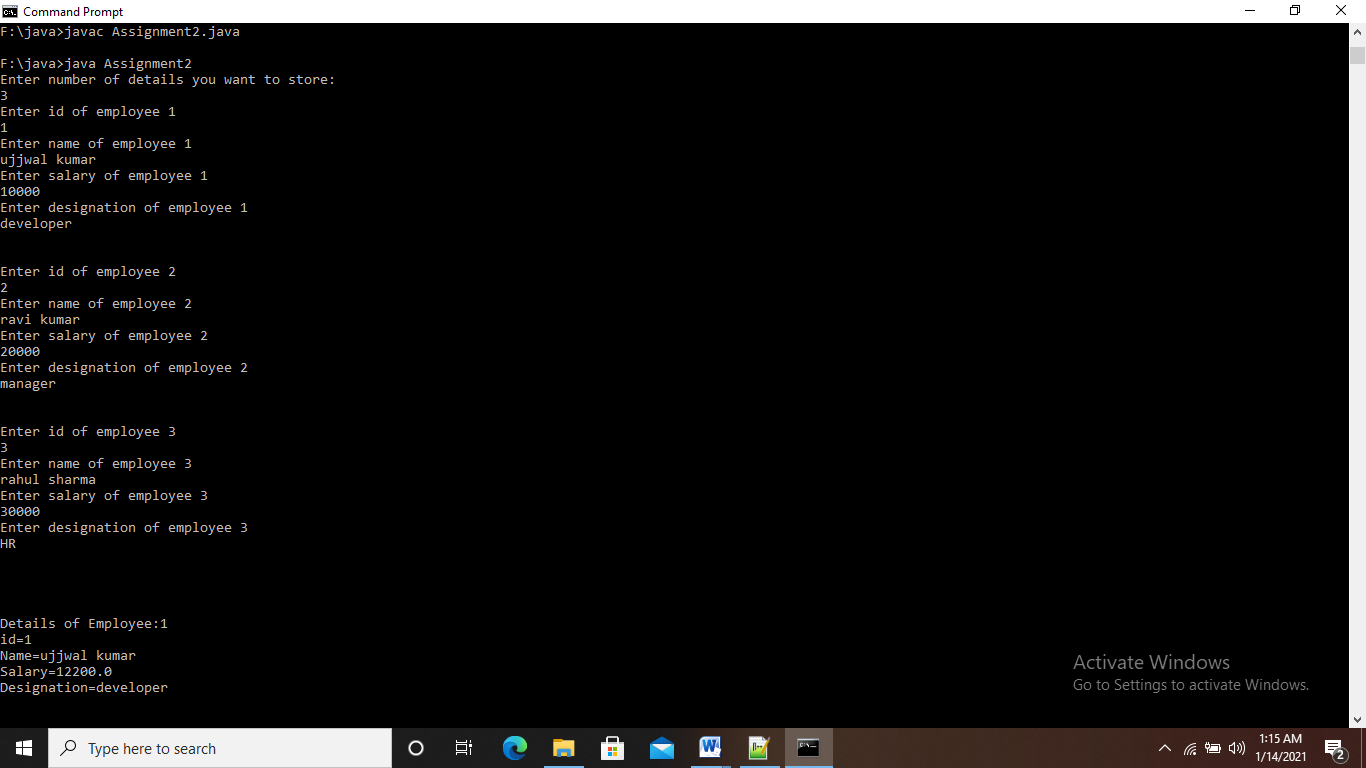
}

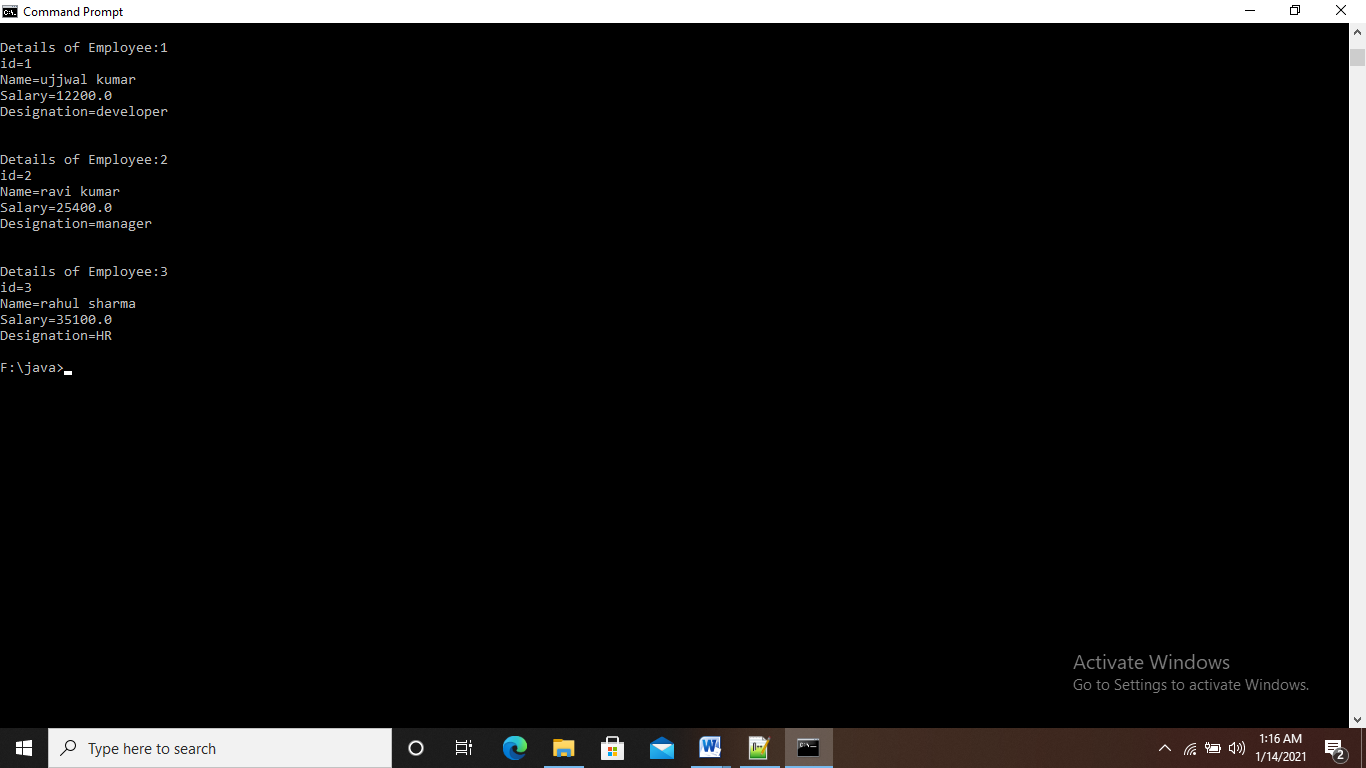
}

}

}

Output:





**Assignment 3**

Create EmployeeDetails class with 4 instance array variables.

EmployeeDetails() : memory size for array id,name,salary, desg must assign in constructor at run time.

read()

read all employee id,name,salary,desg

calSalary()

hra, da, pf local variables.

calculate salary

bonus()

apply bonus

display()

display details

EmployeeTest :

Main methods

Object creation

And calling all methods

Take n number records through keyboards as Id,Name,Salary,Desg

(array id,name,salary,Desg).

Salary = salary + hra + da – pf;

Hra is 10% salary

Da is 7 % salary

Pf 5 % salary

If desg is manager

15% bonus

If developer 10% bonus

Else

5 % bonus

Id, name, salary( grossSalary +bonus ) and desg

**Solution:**

import java.util.Scanner;

class EmployeeDetails

{

Scanner sc=new Scanner(System.in);

int n;

EmployeeDetails(int size)

{

n=size;

id=new int [size];

name=new String [size];

salary=new float [size];

salarycpy=new float[size];

desg=new String [size];

}

int []id;

String []name;

float []salary;

float []salarycpy;

String []desg;

//......................................................................

void read()

{

for(int i=0;i<n;i++)

{

System.out.println("Enter id of employee "+(i+1));

id[i]=sc.nextInt();

sc.nextLine();

System.out.println("Enter name of employee "+(i+1));

name[i]=sc.nextLine();

System.out.println("Enter salary of employee "+(i+1));

salary[i]=sc.nextFloat();

salarycpy[i]=salary[i];

sc.nextLine();

System.out.println("Enter designation of employee "+(i+1));

desg[i]=sc.nextLine();

System.out.println("\n");

}

}

//....................................................................

void calsalary()

{

for(int k=0;k<n;k++)

{

float hra,da,pf;

hra=salary[k]\*0.1f;

da=salary[k]\*0.07f;

pf=salary[k]\*0.05f;

salary[k]=salary[k]+hra+da-pf;

}

}

//.....................................................................

void bonus()

{

for(int k=0;k<n;k++)

{

salary[k]=salarycpy[k];

float hra,da,pf;

hra=salary[k]\*0.1f;

da=salary[k]\*0.07f;

pf=salary[k]\*0.05f;

if(desg[k].equals("manager"))

{

salary[k]=(salary[k]+hra+da-pf)+(salary[k]\*0.15f);

}

else if(desg[k].equals("developer"))

{

salary[k]=(salary[k]+hra+da-pf)+(salary[k]\*0.1f);

}

else

{

salary[k]=(salary[k]+hra+da-pf)+(salary[k]\*0.05f);

}

}

}

//.......................................................................

void display()

{

for(int k=0;k<n;k++)

{

System.out.println("Details of Employee:"+(k+1));

System.out.println("id="+id[k]);

System.out.println("Name="+name[k]);

System.out.println("Salary="+salary[k]);

System.out.println("Designation="+desg[k]);

System.out.println("\n");

}

}

}

//........................................................................

class EmployeeTest

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter Number of records you want to store:");

int x=sc.nextInt();

EmployeeDetails empd=new EmployeeDetails(x);

empd.read();

empd.calsalary();

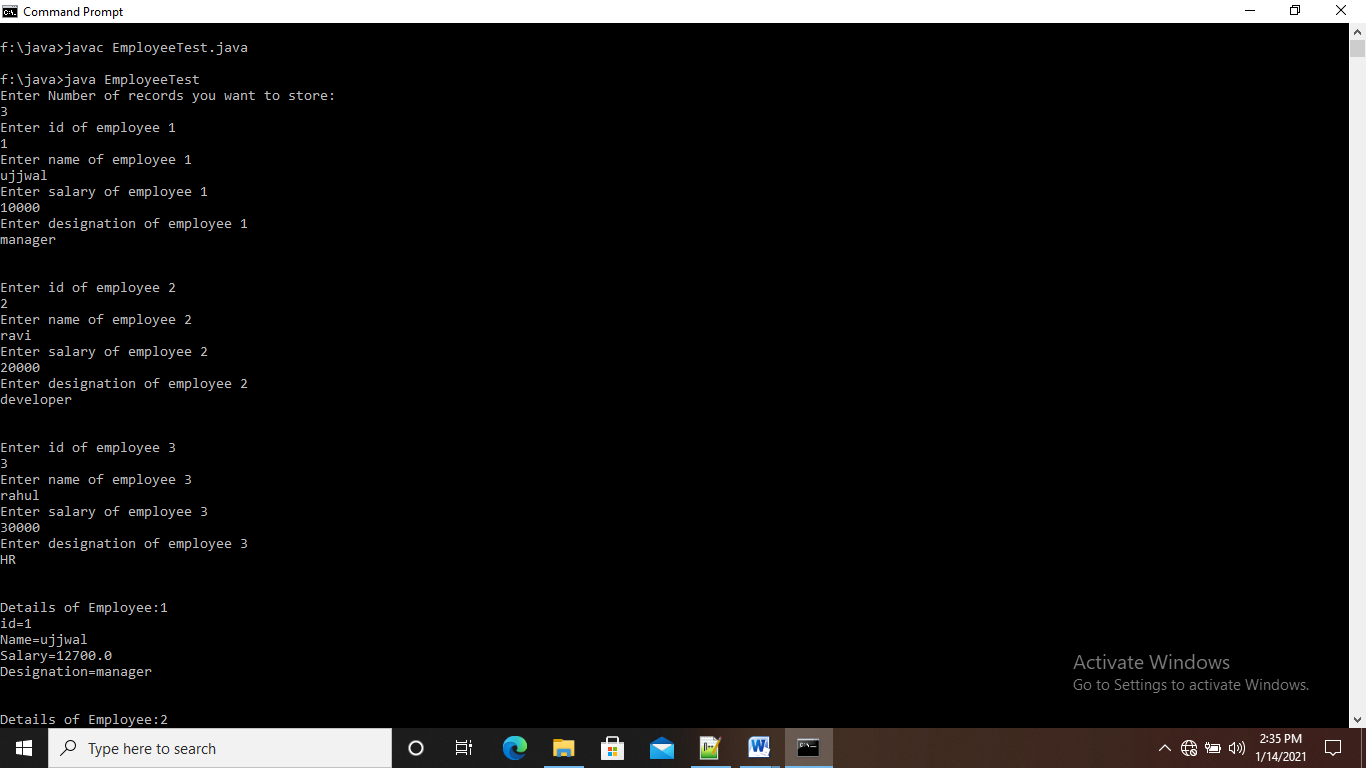
empd.bonus();

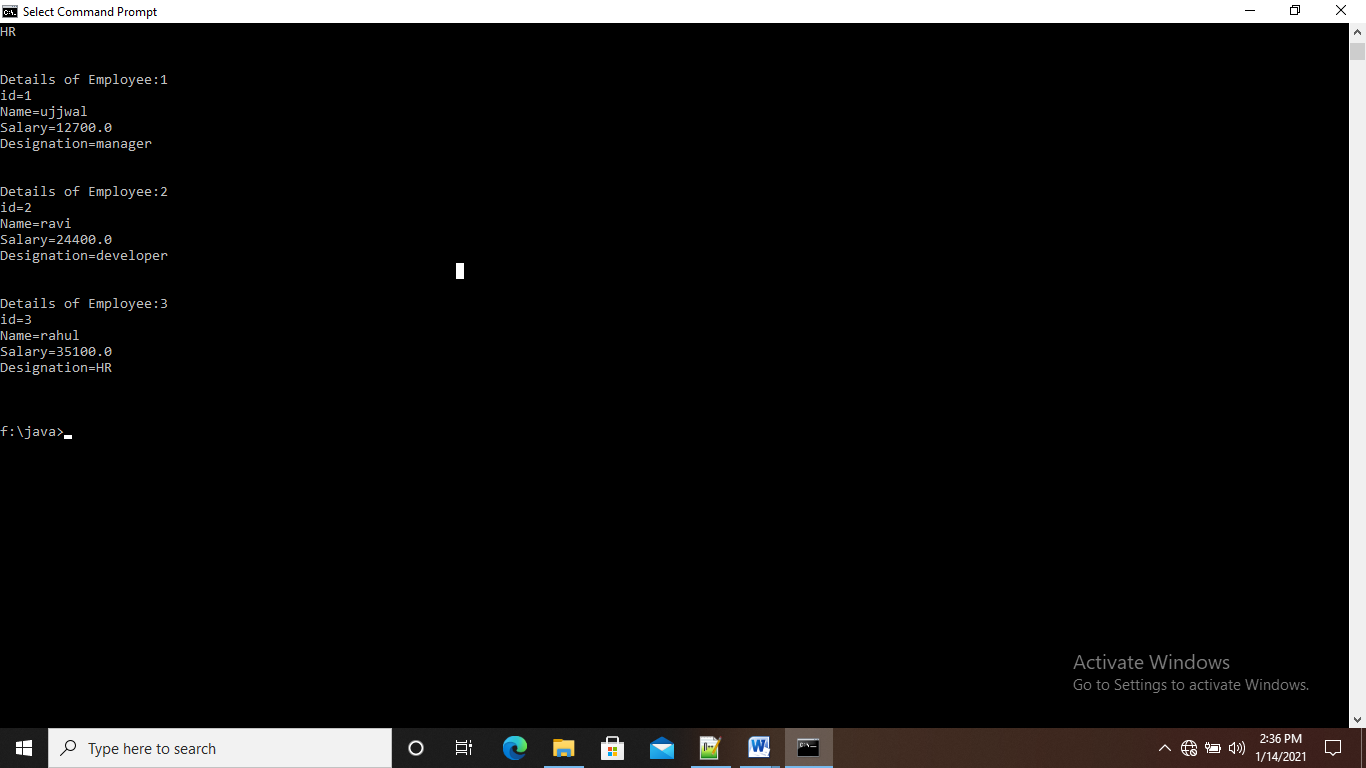
empd.display();

}

}

Output:





**Assignment 1**

do{

Online Examination

1:English , 2 : Math 3 : GK

switch() {

case 1

3 Q

case 2

3 Q

case 3

3 Q

}

Do want to continue ?

}while();

Result g\_total > 70

Result + 10

Result>=90 selected else try next time.

**Solution:**

import java.util.Scanner;

public class OnlineExam

{

public static void main(String args[])

{

System.out.println("Online Examination System");

Scanner sc=new Scanner(System.in);

int choice;

int c=0,d=0,e=0,n=3,marks1=0,marks2=0,marks3=0;

do{

System.out.println("Press 1 for English \nPress 2 for Maths \nPress 3 for GK ");

choice=sc.nextInt();

switch(choice)

{

case 1:if(c==0)

{

System.out.println("Welcome to English Examination");

System.out.println("1.Please,stop.....so many mistakes.");

System.out.println("1. to make\n2. make\n3. making\n4. makes");

int p1=sc.nextInt();

if(p1==3)

{

marks1=marks1+10;

}

else{

System.out.println("Wrong Answer.");

}

System.out.println("2.The English.....English");

System.out.println("1. speak\n2. spoke\n3. spoken\n4. is spoken");

int p2=sc.nextInt();

if(p2==1)

{

marks1=marks1+10;

}

else{

System.out.println("Wrong Answer.");

}

System.out.println("3.The rain comes.....the clouds.");

System.out.println("1. in\n2. near\n3. from\n4. under");

int p3=sc.nextInt();

if(p3==3)

{

marks1=marks1+10;

}

else{

System.out.println("Wrong Answer.");

}

System.out.println("This Section Completed successfully...");

c++;

n--;

break;

}

else{

System.out.println("You have already done this section");

break;

}

case 2:if(d==0)

{

System.out.println("Welcome to Maths Examination");

System.out.println("1.Can we write 0 in the form of p/q?");

System.out.println("1. Yes\n2. No\n3. Cannot be explained\n4. None of the above");

int q1=sc.nextInt();

if(q1==1)

{

marks2=marks2+10;

}

else{

System.out.println("Wrong Answer.");

}

System.out.println("2.Every rational number is:");

System.out.println("1. Whole number\n2. Natural number\n3. Integer\n4. Real number");

int q2=sc.nextInt();

if(q2==4)

{

marks2=marks2+10;

}

else{

System.out.println("Wrong Answer.");

}

System.out.println("3.If the coordinates of a point are(0,-4),then it lies in:");

System.out.println("1. X-axis\n2. Y-axis\n3. At origin\n4. Between x-axis and y-axis");

int q3=sc.nextInt();

if(q3==2)

{

marks2=marks2+10;

}

else{

System.out.println("Wrong Answer.");

}

System.out.println("This Section Completed successfully...");

d++;

n--;

break;

}

else{

System.out.println("You have already done this section");

break;

}

case 3:if(e==0)

{

System.out.println("Welcome to GK Examination");

System.out.println("1.Which is longest river in the world?");

System.out.println("1. Ganga\n2. Nile\n3. Amazon\n4. Niger");

int r1=sc.nextInt();

if(r1==2)

{

marks3=marks3+10;

}

else{

System.out.println("Wrong Answer.");

}

System.out.println("2.Which is the biggest continent in the world?");

System.out.println("1. North America\n2. Aisa\n3. Africa\n4. Australia");

int r2=sc.nextInt();

if(r2==2)

{

marks3=marks3+10;

}

else{

System.out.println("Wrong Answer.");

}

System.out.println("3.Which is india's first super computer?");

System.out.println("1. Param8000\n2. param80000\n3. param800\n4. param8");

int r3=sc.nextInt();

if(r3==1)

{

marks3=marks3+10;

}

else{

System.out.println("Wrong Answer.");

}

System.out.println("This Section Completed successfully...");

e++;

n--;

break;

}

else{

System.out.println("You have already done this section");

break;

}

default:System.out.println("invalied choice...");

}

if(n!=0)

{

System.out.println("Do you want to continue with another section?\nPress 1 for YES \nPress 2 for NO ");

int ch=sc.nextInt();

switch(ch)

{

case 1:System.out.println("YES");

break;

case 2:System.out.println("NO");

if(c==0 || d==0 || e==0)

{

System.out.println("You have not completed the Exam...Please Complete");

}

break;

default:System.out.println("invalied choice...");

}

}

else{

System.out.println("Exam Completed");

System.out.println("Thank You");

}

}while(n!=0);

int result=0;

System.out.println("Marks in english="+marks1);

System.out.println("Marks in maths="+marks2);

System.out.println("Marks in GK="+marks3);

int totalmarks=marks1+marks2+marks3;

System.out.println("Total Marks="+totalmarks);

if(totalmarks>=70)

{

result=totalmarks+10;

System.out.println("Final Marks="+result);

}

if(result>=90)

{

System.out.println("Congratulations. You are Selected");

}

else{

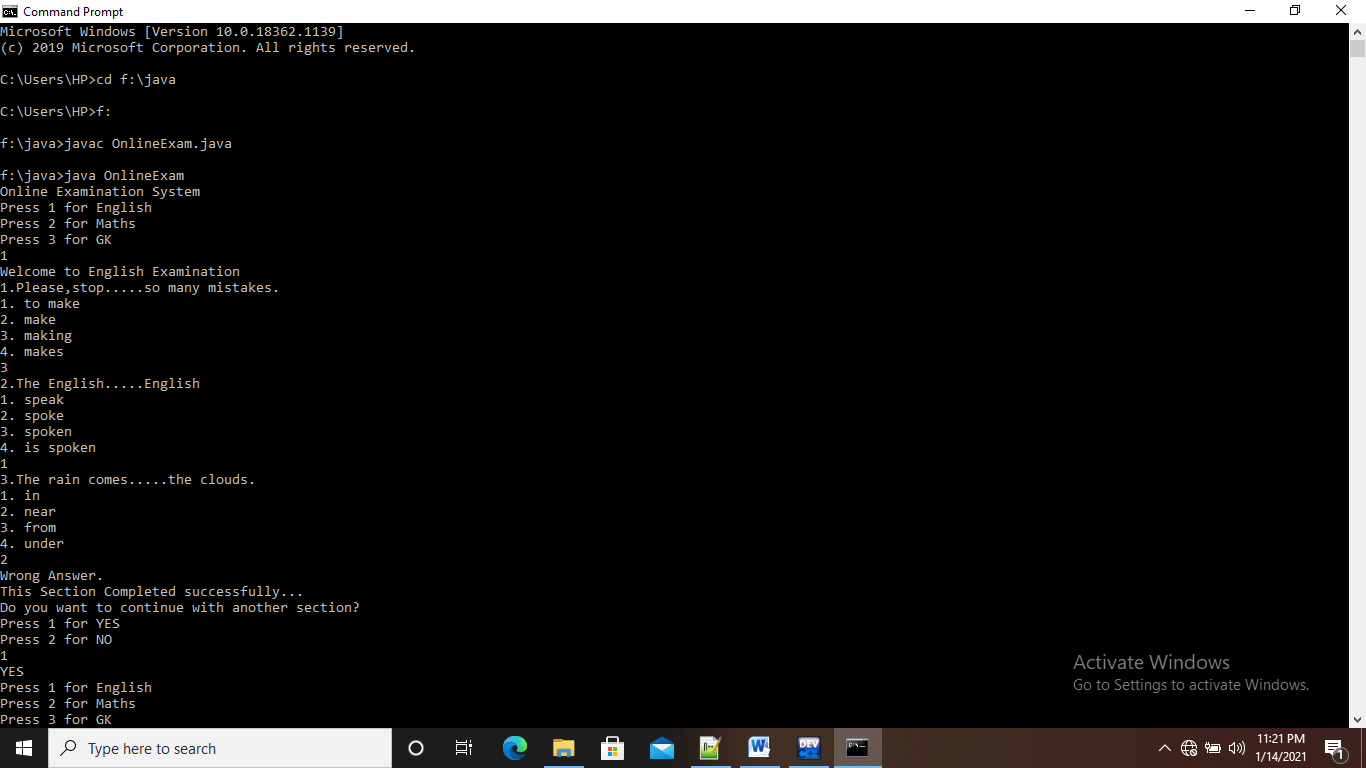
System.out.println("Sorry, not passed. Please try next time");

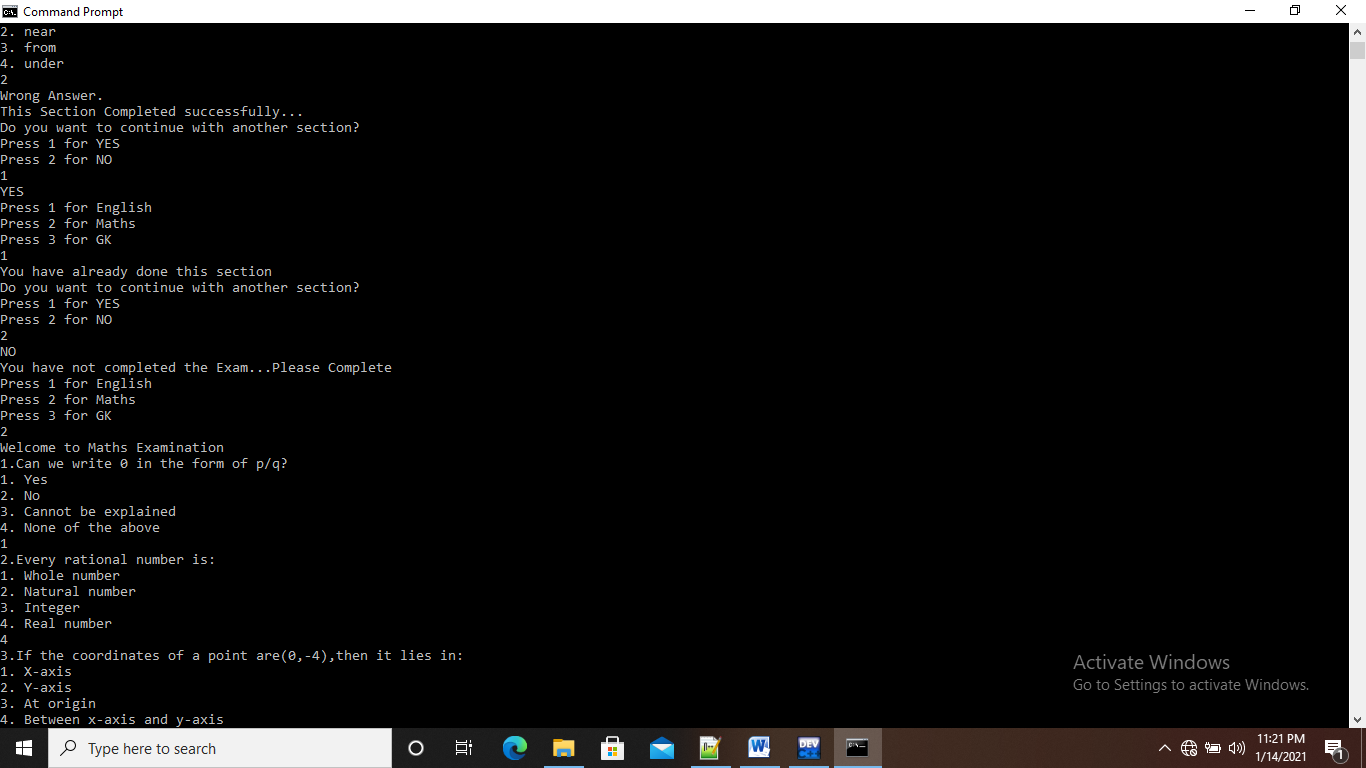
}

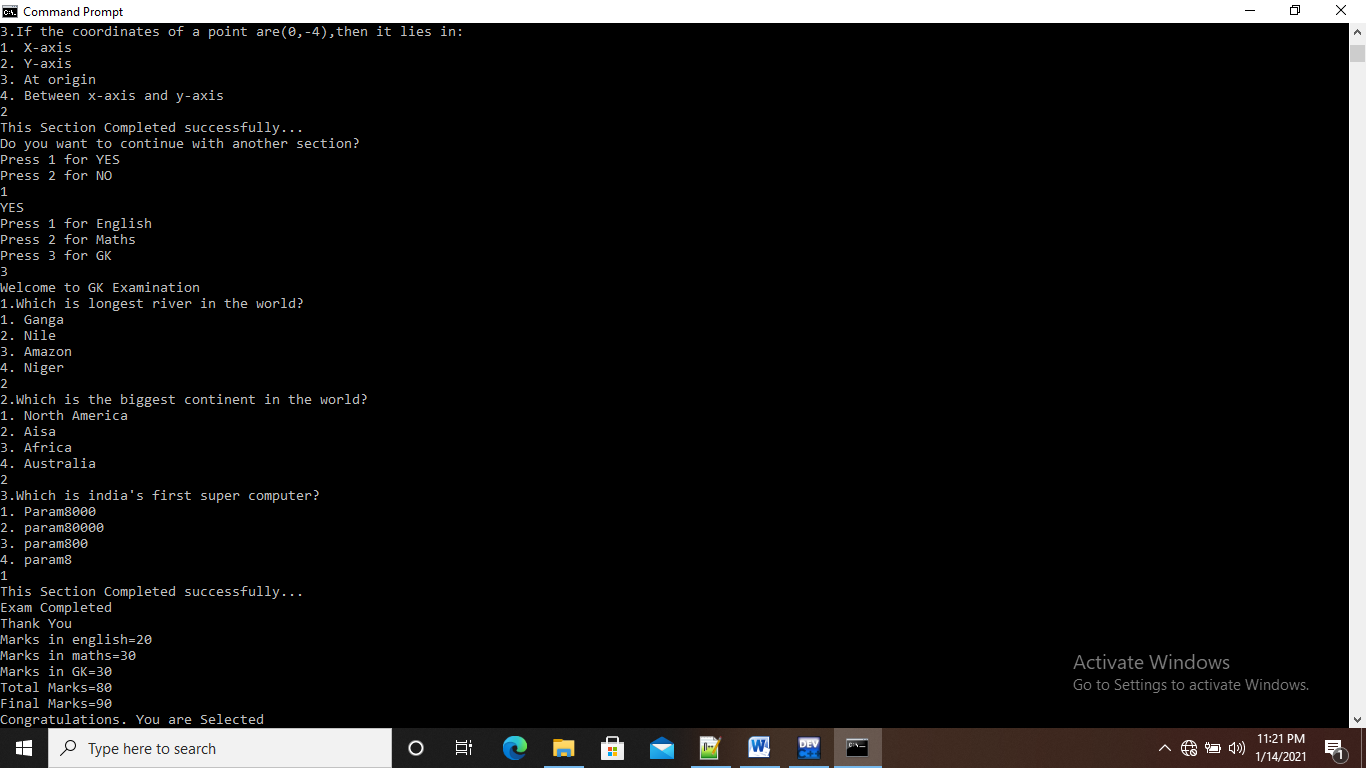
}

}

**Output:**

****

****

****