A

Project Report On

STAFF MANAGEMENT DATABASE IN KMIT

Submitted in partial fulfilment of the requirements for award of the degree of

BACHELOR OF TECHNOLOGY

By

VENKATA GAYATHRI PERI(15BD1A0440)

GUDA SAI RAMYA(15BD1A0428)

Under the guidance of

Lt Col UMESH GOGTE

Faculty

Of

KMIT-HYD

KESHAV MEMORIAL INSTITUE OF TECHNOLOGY
Powered by GENESIS
Accredited by NBA for IT
Approved by AICTE and Affiliated to JNTUH

Contents

ABSTRACT

Our project is based on problems faced by a manager OF STAFF in Kmit. The manager faces problems such as:

- Maintaining hardcopies of employees.
- Knowing how many employees work in each department
- Renewing the records whenever an Employee leaves or gets into KMIT staff
- Updating the records whenever an Employee changes his department
- Calculation of salary departure for a particular department

In order for these problems to be solved, all employees and departments will be registered on data base, salary of a particular department, list of employees in it, adding an employee, deleting an employee and updating details and listing of them with ease would be possible.

Scope of the project

This project is handy to keep track of all the employees in the organisation, it keeps track of all the employees in organisation and department vice track. The effort and time taken to maintain the records is significantly reduced when compared to the manual procedure of record keeping.

The project performs following operations:

1. Insertion of an new employee
2. Insertion of a new department
3. Deleting an employee when he/she leaves the organization
4. Deleting an department when it is terminated
5. Updating the name of an employee
6. Updating the department of an employee
7. Information about the departments(number of employees, salary departure for department)
8. It provides us with total available information about an employee on providing his employee_id
9. It Keeps the track of all Employees in a organization and also department vice list of employees
10. Future scope of project
    10.1 salary increments
    10.2 over Time Calculations
        can be added to project for to make it more convenient for the Organisation

Glossary

The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development.

MySQL (My structured Query language): relational database management system that runs as a server providing multi-user access to number of databases.

Computer with Operating System Windows 7 or above

Notepad++ is a text editor and source code editor for use with Microsoft Windows. It supports tabbed editing, which allows working with multiple open files in a single window.

Data Base

4.1: Creating a database

Use the command **CREATE database_name; in mysql** command line client to create a database

4.2: Tables in Data Base

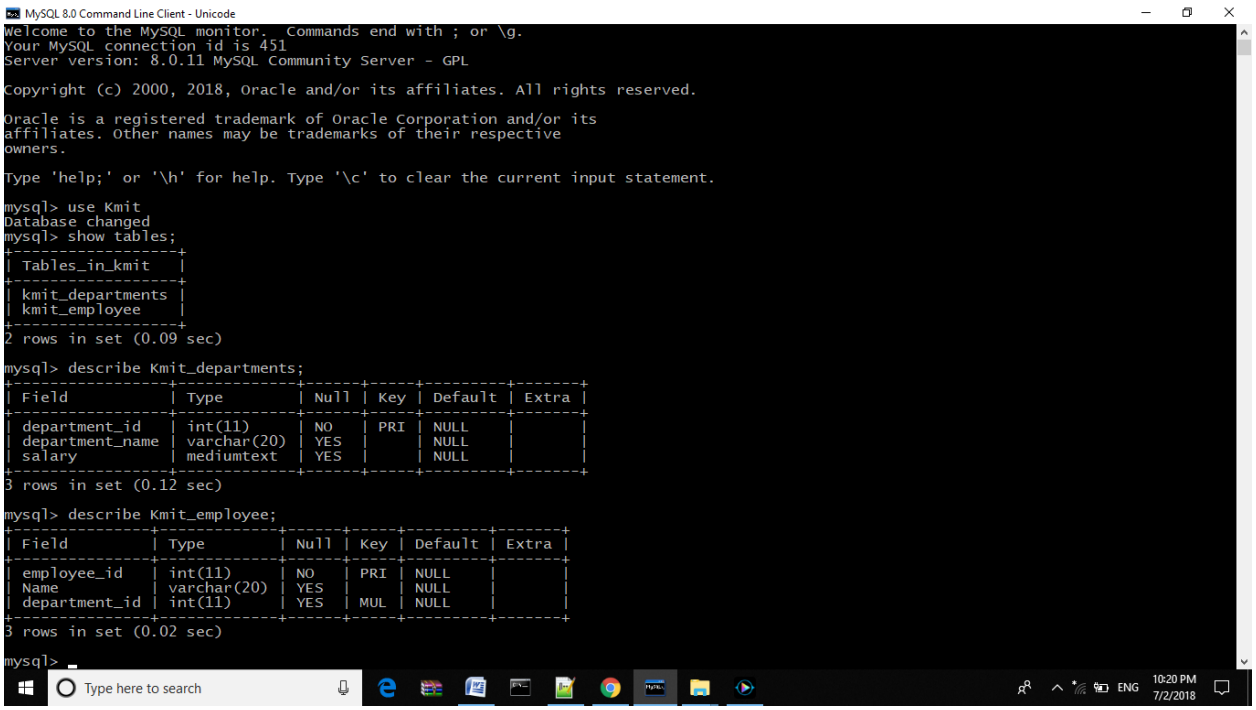
Create two tables in database employee and department

Use the command **Create Table TableName (FeildName DataType);**

Set the department_id as primary key in department table

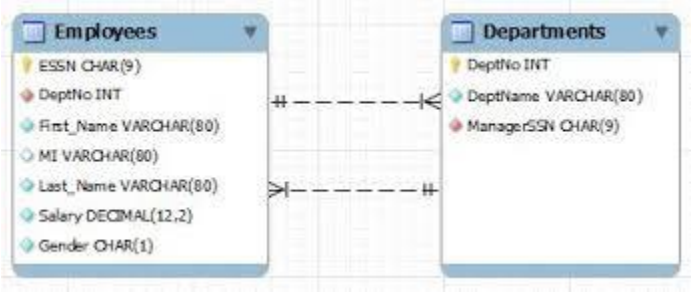
Set the employee_id as primary key and department_id as a foreign key with reference to department table in employee table

4.3: Description of Tables



4.4: Relation between Department and employee table

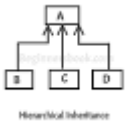
There is a 1 to n relationship between the department and employee Table



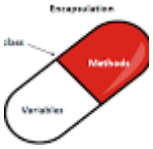
Java Code

5.1 Basic Oops Concepts

1. Inheritance



2. Encapsulation



3. Abstraction

4. Polymorphism

5.2 Classes in Code

1. Kmit

2. Insert

3. Delete

4. Update

5. InfoAboutDepartment

6. listOfDepartments

7. listOfEmployees

8. Logo

Kmit

Kmit is the main class in the program. It is the parent class for all other classes. It contains only one method the static main method. It works on Switch statement. It calls the respective methods on entering an int key. Connection to database is established in this class and terminated in this class.

Code of Kmit Class

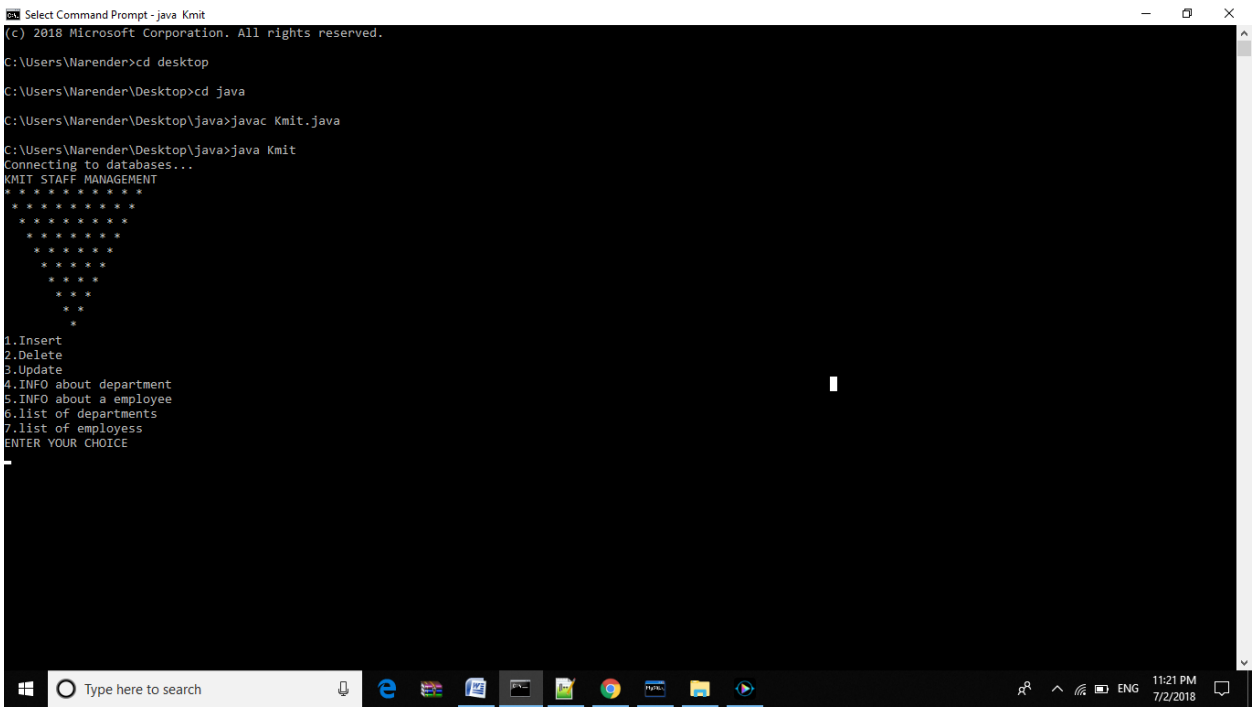
```
/*
import java.sql.*;
import java.util.*;
class Kmit
{
        //JDBC driver name and databse url
   static final String JDBC_DRIVER="com.mysql.cj.jdbc.Driver";
   static final String DB_URL="jdbc:mysql://localhost/Kmit?useSSL=false";
   // DataBASe credinals
   static final String USER="root";
   static final String PASS="ramya@123";
        static Connection conn=null;
        static PreparedStatement stmt=null;
        static ResultSet rs=null;

   public static void main(String[] args)
        {

                try
                {
                  Class.forName("com.mysql.cj.jdbc.Driver");
                  conn=DriverManager.getConnection(DB_URL,USER,PASS);
                  System.out.println("Connecting to databases...");
                  Logo l=new Logo();
                  Scanner sc=new Scanner(System.in);
                  char ch;
                  do
                  {
                     System.out.println("1.Insert\n2.Delete\n3.Update\n4.INFOabout
department\n5.INFO about a employee\n6.list of departments");
                           System.out.println("7.list of employess");
                     System.out.println("ENTER YOUR CHOICE");
                     switch(sc.nextInt())
                     {
                           case 1:
```

```
                System.out.println("1.Insert a employee into a dept\n2.Insert a
department into kmit");
                System.out.println("ENTER YOUR CHOICE");
                Insert i=new Insert();
                switch(sc.nextInt())
                {
                        case 1:
                        System.out.println("please enter the
employee_id,Name,Department_id ");
              i.insertinemp(sc.nextInt(),sc.next(),sc.nextInt());
                        break;
                        case 2:
                        System.out.println("enter the values
department_id,department_name,salaray ");
              i.insertindept(sc.nextInt(),sc.next(),sc.nextInt());
                        break;
                        default :
              System.out.println("please enter a valid option");
                 break;
                }
            break;
           case 2:
                System.out.println("1.delete a employee from a dept\n2.delete a
department from kmit");
                System.out.println("ENTER YOUR CHOICE");
                Delete d=new Delete();
                switch(sc.nextInt())
                {
                        case 1:
                        System.out.println("enter the employee_id");
              d.deleteEmployee(sc.nextInt());
                        break;
                        case 2:
                        System.out.println("please enter Department_id ");
              d.deleteDepartment(sc.nextInt());
                        break;
                        default :
              System.out.println("please enter a valid option");
                 break;

                }
            break;
               case 3:
               Update u=new Update();
               System.out.println("1.update name\n2.update department");
```

```
                    switch(sc.nextInt())
                    {
                        case 1:
                        System.out.println("enter the id number of employee whose name you want to chane");
                        int x=sc.nextInt();
                        System.out.println("enter the new name");
                        String y=sc.next();
                        u.updateName(x,y);
                        break;
                        case 2:
                        System.out.println("enter the id number of employee whose name you want to chane");
                        int a=sc.nextInt();
                        System.out.println("enter the new department_id");
                        int b=sc.nextInt();
                        u.updateDepartment(a,b);
                        break;
                        default :
                    System.out.println("please enter a valid option");
                        break;
                    }

                    break;
                case 4:
                      System.out.println("1.department_id of a department\n2.number of employes in a department\n3.salary departure of a department");
                            System.out.println("ENTER YOUR CHOICE");
                            InfoAboutDepartment iad=new InfoAboutDepartment();
                            switch(sc.nextInt())
                            {
                                case 1:
                                System.out.println("enter name of the department whose department you wnt to know:");
                                String s=sc.next();
                                iad.getDepartment(s);
                                break;
                                case 2:
                                System.out.println("enter the department_id of a department to know the number of employees in it");
                                iad.getEmployees(sc.nextInt());
                                break;
                                case 3:
                                System.out.println("enter the department_id of a department to know the salary departure of department");
```

```
                                        iad.salarayDeparture(sc.nextInt());
                                        break;
                                        default :
                                System.out.println("please enter a valid option");
                                    break;

                                        }
                                break;
                            case 5:
                                System.out.println("enter employee id");
                                InfoAboutEmployee iae=new InfoAboutEmployee(sc.nextInt());
                                break;
                                case 6:
                                listOfDepartments lod=new listOfDepartments();
                                lod.departmentsInKmit();
                            break;
                                case 7:
                                System.out.println("1.total employess in KMit\n2.List of
employees in a department");
                                listOfEmployees loe=new listOfEmployees();
                                switch(sc.nextInt())
                                {
                                        case 1:
                                        loe.listofemplloyees();
                                        break;
                                        case 2:
                                    System.out.println("enter department_id for the list of employes
in a department");
                                    loe.employeesInDepartment(sc.nextInt());
                                        break;
                                        default :
                                System.out.println("please enter a valid option");
                                    break;
                                }
                                break;
                            default :
                            System.out.println("please enter a valid option");
                                break;
                    }
                        System.out.println("Do you want to continue .?(y/n) ");
                        ch=sc.next().charAt(0);
                }while(ch=='y' ||ch=='Y');

            }
            catch(SQLException se)
```

```
        {
                System.out.println("connecting to database failed");
                se.printStackTrace();
         }
              catch(InputMismatchException ie)
              {
                      System.out.println("NOTE : ");
                      System.out.println("please enter the valid details");
                      System.out.println("enter according to above mentoined sequence");
                      System.out.println("there are no spaces between Name");
              }
    catch(Exception e)
        {
                e.printStackTrace();
        }
              finally
        {
                //fianlly block used to close resources
                try{
                      if(stmt!=null)
                            stmt.close();
                }catch(SQLException se2){
                      //nothing we can do
                }
                try{
                      if(conn!=null)
                            conn.close();
                }catch(SQLException se){
                      se.printStackTrace();
                }

        }

              System.out.println("ended..!");
      }


}
*/
```


Select Command Prompt - java Kmit
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\Narender>cd desktop
C:\Users\Narender\Desktop>cd java
C:\Users\Narender\Desktop\java>javac Kmit.java
C:\Users\Narender\Desktop\java>java Kmit
Connecting to databases...
KMIT STAFF MANAGEMENT
1.Insert
2.Delete
3.Update
4.INFO about department
5.INFO about a employee
6.list of departments
7.list of employess
ENTER YOUR CHOICE
Type here to search
ENG
11:21 PM
7/2/2018

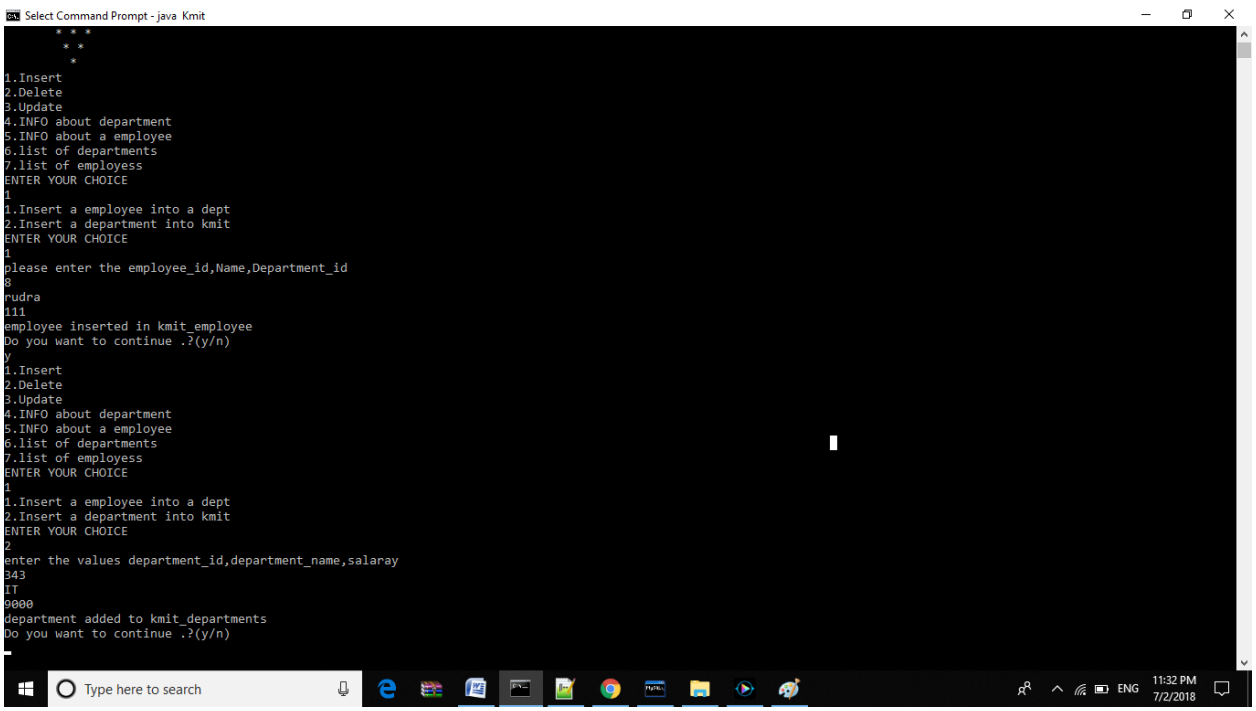
2. Insert

The insert class is used to insert a new employee or department into kmit. It is called when the user enter 1 as the input. It has two methods insertindept and insertinemp which insert new department into kmit and new employee to a department respectively when called by passing appropriate parameters. Two method are called in a switch case

Code of Insert Class

```
/*
class Insert extends Kmit
{
        public void insertindept(int n,String deptname,int s)
        {
         try
         {
                String str="INSERT INTO
Kmit_departments(department_id,department_name,salary)"+"VALUES(?,?,?);";
          stmt=conn.prepareStatement(str);
                stmt.setInt(1,n);
                stmt.setString(2,deptname);
                stmt.setInt(3,s);
          stmt.execute();
                System.out.println("department added to kmit_departments");
         }
         catch(SQLIntegrityConstraintViolationException se1)
         {
                 System.out.println("This Id already exists please enter another Id");
         }
         catch(SQLException se)
         {
                //handling errors of jdbc
                se.printStackTrace();
         }
    catch(Exception e)
         {
                //handling errors of jdbc
                e.printStackTrace();
         }
        }
        public void insertinemp(int n,String empname,int depid)
        {
         try
         {
```

```
                String                          str="INSERT                          INTO
Kmit_employee(employee_id,Name,department_id)"+"VALUES(?,?,?)";
                stmt=conn.prepareStatement(str);
                stmt.setInt(1,n);
                stmt.setString(2,empname);
                stmt.setInt(3,depid);
              stmt.execute();
                System.out.println("employee inserted in kmit_employee");
            }
            catch(SQLIntegrityConstraintViolationException se1)
            {
                  System.out.println("please enter a valid Id");
            }
            catch(SQLException se)
            {
                  //handling errors of jdbc
                  se.printStackTrace();
            }
      catch(Exception e)
            {
                  //handling errors of jdbc
                  e.printStackTrace();
            }
          }
}
*/
```



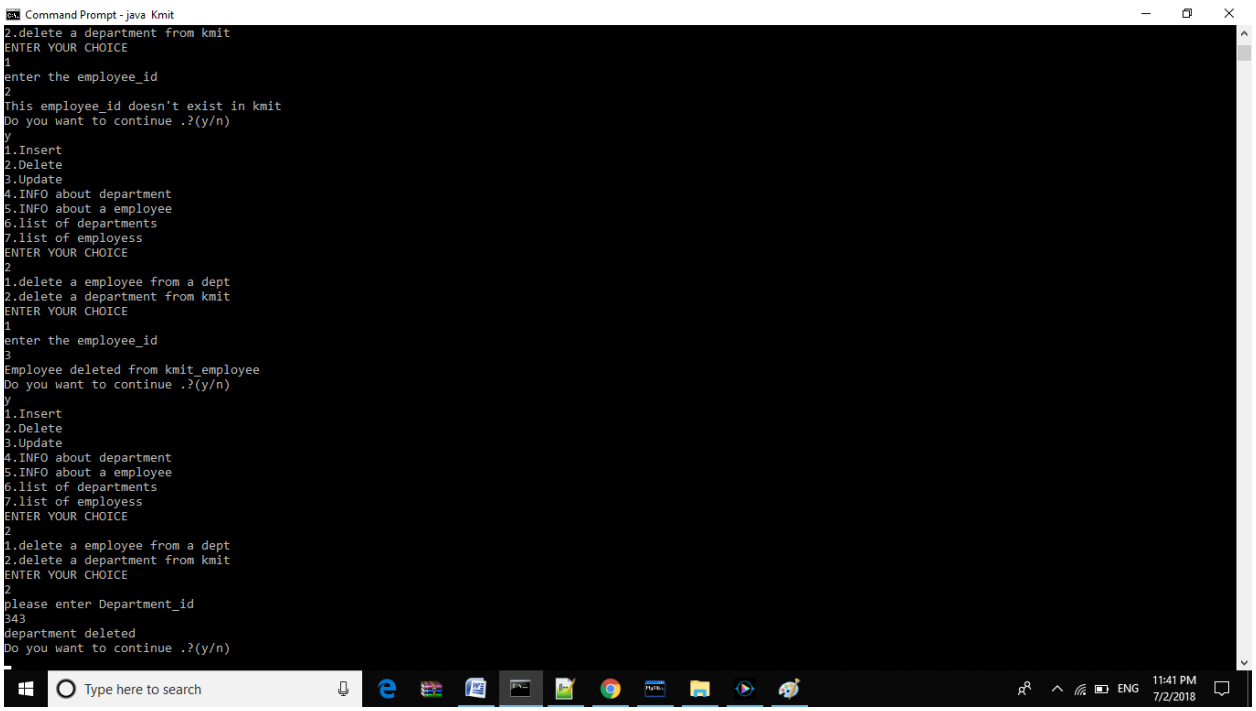
3. Delete

The Delete class is used to delete a employee or department into kmit. It is called when the user enter 2 as the input. It has two methods deleteDepartment and deleteEmployee which delete's a department from kmit and employee from department respectively when called by passing appropriate parameters. Two method are called in a switch case

Code of Delete Class

```
/*
class Delete extends Kmit
{
	public void deleteDepartment(int n)
	{
		int y=0;
		try
		{
			String str2="SELECT count(*) FROM Kmit_departments WHERE department_id ='"+n+"';";
			stmt=conn.prepareStatement(str2);
			rs=stmt.executeQuery();
			while(rs.next())
			{
	   y=rs.getInt(1);
	}
			if(y==0)
				System.out.println("this epartment_id doesn't exist in kmit");
			else
			{
			   String str="delete from Kmit_employee where department_id=?;";
		stmt=conn.prepareStatement(str);
		stmt.setInt(1,n);
			   stmt.execute();
			   String str1="delete from Kmit_departments where department_id=?;";
			   PreparedStatement stmt1=conn.prepareStatement(str1);
		stmt1.setInt(1,n);
			   stmt1.execute();
				System.out.println("department deleted");
			}

		}
		catch(SQLException se)
	   {
		 //handling errors of jdbc
		 se.printStackTrace();
```

```
            }
        catch(Exception e)
            {
                    //handling errors of jdbc
                    e.printStackTrace();
            }

          }
          public void deleteEmployee(int n)
          {
                  int y=0;
                  try
                  {
                          String str1="SELECT count(*) FROM Kmit_employee WHERE
employee_id ='"+n+"';";
                          stmt=conn.prepareStatement(str1);
                          rs=stmt.executeQuery();
                          while(rs.next())
                          {
             y=rs.getInt(1);
          }
                          if(y==0)
                                  System.out.println("This employee_id doesn't exist in kmit");
                          else
                          {
                            String str="delete from Kmit_employee where employee_id=?;";
                        stmt=conn.prepareStatement(str);
                        stmt.setInt(1,n);
                            stmt.execute();
                            System.out.println("Employee deleted from kmit_employee");
                          }
                  }
             catch(SQLException se)
             {
                    //handling errors of jdbc
                    se.printStackTrace();
             }
        catch(Exception e)
             {
                    //handling errors of jdbc
                    e.printStackTrace();
             }
          }
}
*/
```


Command Prompt - java Kmit
2.delete a department from kmit
ENTER YOUR CHOICE
1
enter the employee_id
2
This employee_id doesn't exist in kmit
Do you want to continue .?(y/n)
y
1.Insert
2.Delete
3.Update
4.INFO about department
5.INFO about a employee
6.list of departments
7.list of employess
ENTER YOUR CHOICE
2
1.delete a employee from a dept
2.delete a department from kmit
ENTER YOUR CHOICE
1
enter the employee_id
3
Employee deleted from kmit_employee
Do you want to continue .?(y/n)
y
1.Insert
2.Delete
3.Update
4.INFO about department
5.INFO about a employee
6.list of departments
7.list of employess
ENTER YOUR CHOICE
2
1.delete a employee from a dept
2.delete a department from kmit
ENTER YOUR CHOICE
2
please enter Department_id
343
department deleted
Do you want to continue .?(y/n)
Type here to search
ENG
11:41 PM
7/2/2018

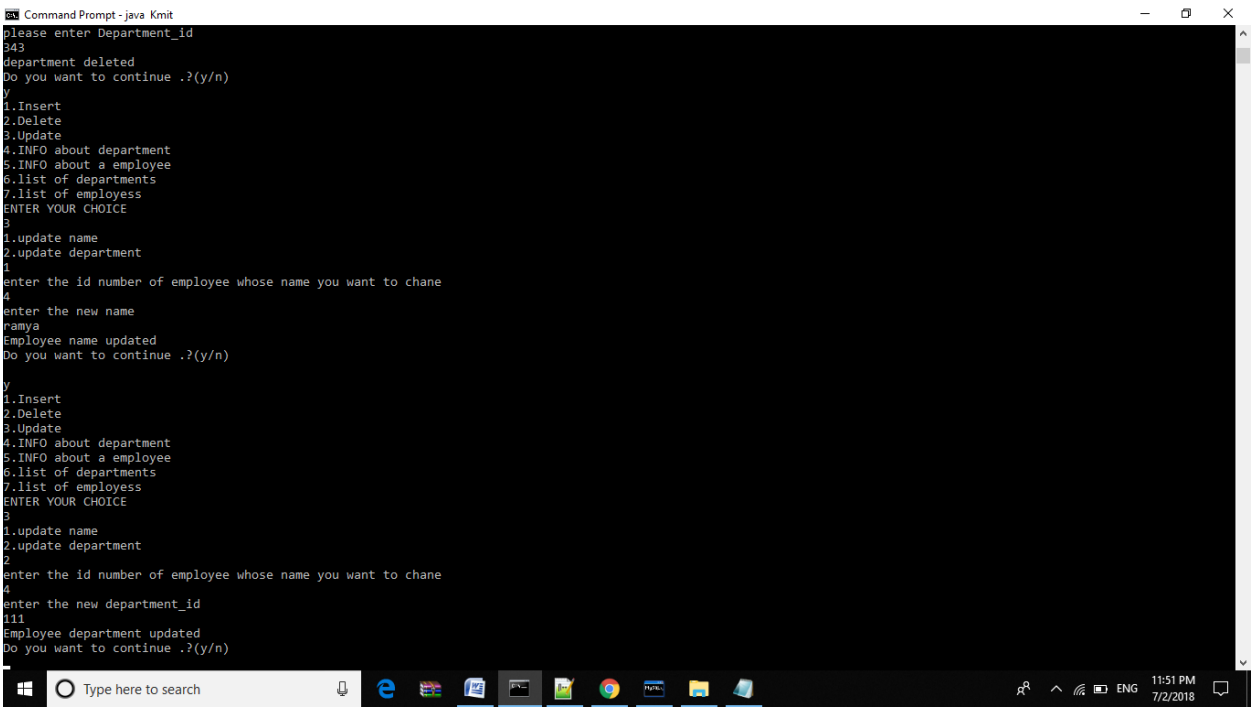
4. Update

The update class is used to update employee's Name or department . It is called when the user enter 3 as the input. It has two methods updateName and updateDepartment which update employee's Name and department respectively when called by passing appropriate parameters. Two method are called in a switch case

Code of Update Class

```
class Update extends Kmit
{
    public void updateName(int n,String s)
    {
            int y=0;
            try
            {
                    String str1="SELECT count(*) FROM Kmit_employee WHERE employee_id ='"+n+"';";
                    stmt=conn.prepareStatement(str1);
                    rs=stmt.executeQuery();
                    while(rs.next())
                    {
          y=rs.getInt(1);
       }
                    if(y==0)
                            System.out.println("This employee_id doesn't exist in kmit");
                    else
                    {
                      String str="update Kmit_employee set Name='"+s+"' where employee_id='"+n+"';";
                stmt=conn.prepareStatement(str);
                      stmt.execute();
                      System.out.println("Employee name updated");
                    }
            }
      catch(SQLException se)
      {
              //handling errors of jdbc
              se.printStackTrace();
      }
     catch(Exception e)
      {
              //handling errors of jdbc
              e.printStackTrace();
      }
    }
```

```
    public void updateDepartment(int n1,int n2)
    {
            int y=0;
            try
            {
                    String str1="SELECT count(*) FROM Kmit_employee WHERE
employee_id ='"+n1+"';";
                    stmt=conn.prepareStatement(str1);
                    rs=stmt.executeQuery();
                    while(rs.next())
                    {
           y=rs.getInt(1);
         }
                    if(y==0)
                            System.out.println("This employee_id doesn't exist in kmit");
                    else
                    {
                      String str="update Kmit_employee set department_id='"+n2+"' where
employee_id='"+n1+"';";
                  stmt=conn.prepareStatement(str);
                      stmt.execute();
                      System.out.println("Employee department updated");
                    }
            }
        catch(SQLException se)
        {
                //handling errors of jdbc
                se.printStackTrace();
        }
      catch(Exception e)
        {
                //handling errors of jdbc
                e.printStackTrace();
        }
    }


}
*/
```


Command Prompt - java Kmit
please enter Department_id
343
department deleted
Do you want to continue .?(y/n)
y
1.Insert
2.Delete
3.Update
4.INFO about department
5.INFO about a employee
6.list of departments
7.list of employess
ENTER YOUR CHOICE
3
1.update name
2.update department
1
enter the id number of employee whose name you want to chane
4
enter the new name
ramya
Employee name updated
Do you want to continue .?(y/n)
y
1.Insert
2.Delete
3.Update
4.INFO about department
5.INFO about a employee
6.list of departments
7.list of employess
ENTER YOUR CHOICE
3
1.update name
2.update department
2
enter the id number of employee whose name you want to chane
4
enter the new department_id
111
Employee department updated
Do you want to continue .?(y/n)
Type here to search
ENG
11:51 PM
7/2/2018

5. InfoAboutDepartment

The InfoAboutDepartment class provides the information about a department. It is called when the user enter 4 as the input. It has three methods getDepartment, getEmployees and salarayDeparture which returns department_id,number of employees in a department and salary departure for particular department respectively when called by passing appropriate parameters. Three method are called in a switch case

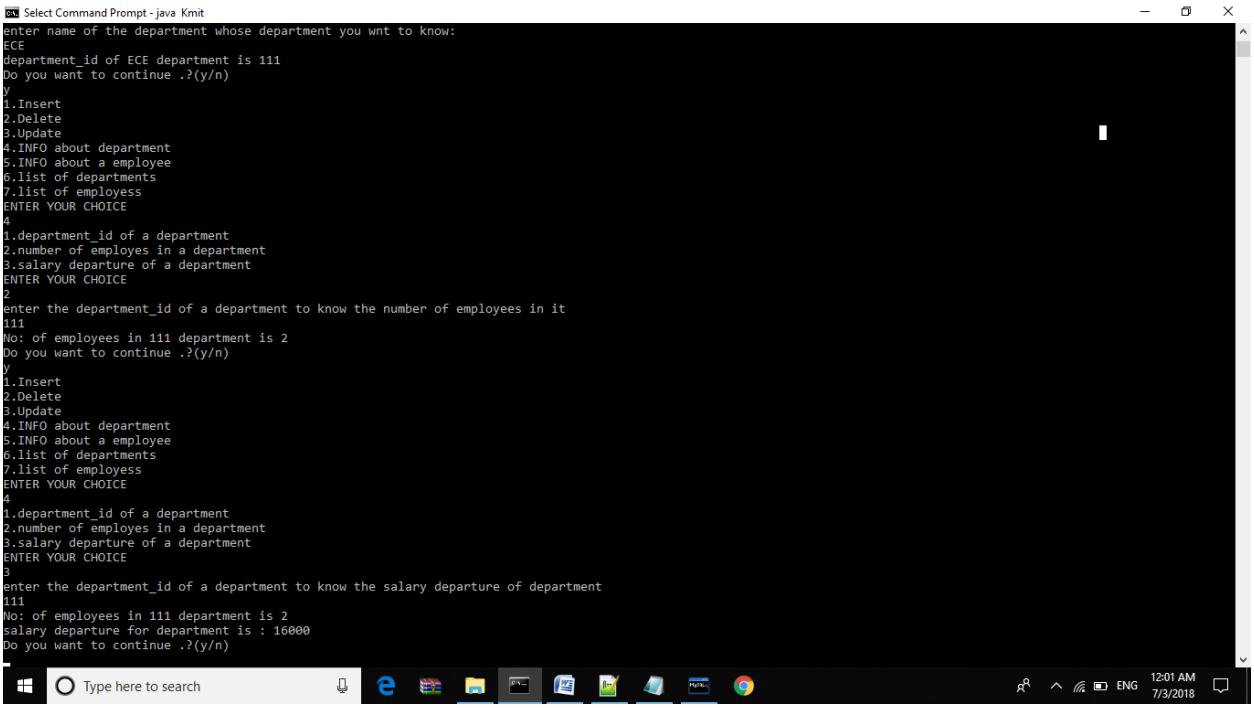
Code of InfoAboutDepartment Class

```
/*  class InfoAboutDepartment extends Kmit
{

    public void getDepartment(String s1)
    {

            int y=0;
            try
            {
              String str1="SELECT count(*) FROM Kmit_departments WHERE department_name ='"+s1+"';";
                    stmt=conn.prepareStatement(str1);
                    rs=stmt.executeQuery();
                    while(rs.next())
                    {
          y=rs.getInt(1);
        }
                    if(y==0)
                            System.out.println("No department exists with this name");
                    else
                    {
                String str="Select department_id from Kmit_departments where department_name=?;";
          stmt=conn.prepareStatement(str);
                stmt.setString(1,s1);
                rs=stmt.executeQuery();
                while(rs.next())
                     {
          System.out.println("department_id of "+s1+" department is "+rs.getInt(1));
         }
              }
            }

            catch(SQLException se)
      {
```

```
            //handling errors of jdbc
            se.printStackTrace();
        }
       catch(Exception e)
        {
            //handling errors of jdbc
            e.printStackTrace();
        }
    }
    public int getEmployees(int n)
    {
            int y=0;
            int v=0;
            try
            {
                    String str1="SELECT count(*) FROM Kmit_departments WHERE department_id ='"+n+"';";
                    stmt=conn.prepareStatement(str1);
                    rs=stmt.executeQuery();
                    while(rs.next())
                    {
          y=rs.getInt(1);
         }
                    if(y==0)
                            System.out.println("No department exists with this id");
                    else
                    {
                      String str="select count(*) from Kmit.Kmit_employee where department_id='"+n+"'";
                stmt=conn.prepareStatement(str);
                     rs=stmt.executeQuery(str);
                    while(rs.next())
                    {
                            v= rs.getInt(1);
                    }
                    System.out.println("No: of employees in "+n+" department is "+v);
                    }
            }
            catch(SQLException se)
      {
            se.printStackTrace();
      }
      catch(Exception e)
        {
            e.printStackTrace();
```

```
        }
            return v;
    }
    public long salarayDeparture(int n)
    {
            int y=0;
            int k=0,sal=0;
            long sum=0;
            try
            {
                    String str1="SELECT count(*) FROM Kmit_departments WHERE
department_id ='"+n+"';";
                    stmt=conn.prepareStatement(str1);
                    rs=stmt.executeQuery();
                    while(rs.next())
                    {
           y=rs.getInt(1);
        }
                    if(y==0)
                            System.out.println("No department exists with this id");
                    else
                    {
                      InfoAboutDepartment iad1=new InfoAboutDepartment();
                      k=iad1.getEmployees(n);
                      String str="SELECT salary from Kmit.Kmit_departments where
department_id='"+n+"';";
                      stmt=conn.prepareStatement(str);
                      rs=stmt.executeQuery();
                      while(rs.next())
                      {
                              sal=rs.getInt(1);
                      }
                      sum=k*sal;
                     System.out.println("salary departure for department is : "+sum);
                    }

            }
            catch(SQLException se)
      {
              se.printStackTrace();
      }
     catch(Exception e)
      {
              e.printStackTrace();
      }
```

```
            return sum;
        }
}
*/
```


Select Command Prompt - java Kmit
enter name of the department whose department you wnt to know:
ECE
department_id of ECE department is 111
Do you want to continue .?(y/n)
y
1.Insert
2.Delete
3.Update
4.INFO about department
5.INFO about a employee
6.list of departments
7.list of employess
ENTER YOUR CHOICE
4
1.department_id of a department
2.number of employes in a department
3.salary departure of a department
ENTER YOUR CHOICE
2
enter the department_id of a department to know the number of employees in it
111
No: of employees in 111 department is 2
Do you want to continue .?(y/n)
y
1.Insert
2.Delete
3.Update
4.INFO about department
5.INFO about a employee
6.list of departments
7.list of employess
ENTER YOUR CHOICE
4
1.department_id of a department
2.number of employes in a department
3.salary departure of a department
ENTER YOUR CHOICE
3
enter the department_id of a department to know the salary departure of department
111
No: of employees in 111 department is 2
salary departure for department is : 16000
Do you want to continue .?(y/n)
Type here to search
ENG
12:01 AM
7/3/2018

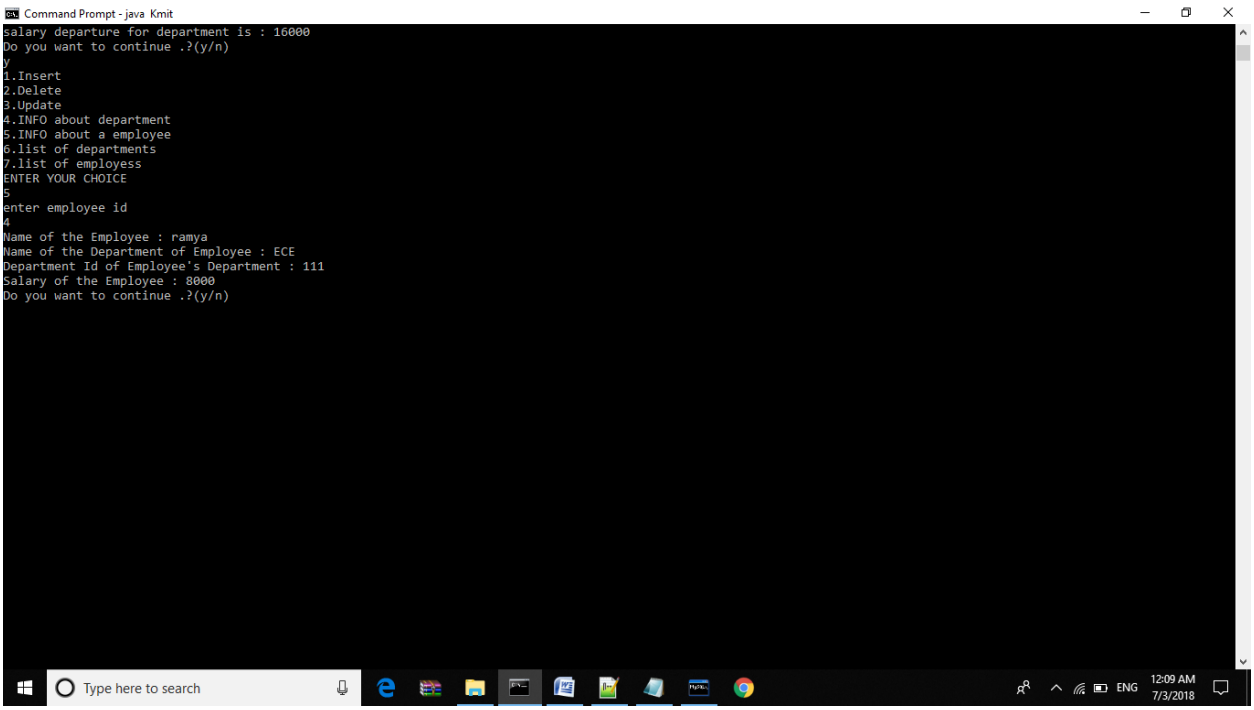
6. InfoAboutEmployee

The InfoAboutEmployee class provides the information about a Employee. It is called when the user enter 5 as the input. It has a parameterized constructer which when called by passing appropriate parameters provides us with all details of employee related to passed parameter.

Code of InfoAboutEmployee Class

```
/*
class InfoAboutEmployee extends Kmit
{
   InfoAboutEmployee(int n)
     {
              int y=0;
       try
              {
                 String str1="SELECT count(*) FROM Kmit_employee WHERE employee_id
="'+n+"';";
                        stmt=conn.prepareStatement(str1);
                        rs=stmt.executeQuery();
                        while(rs.next())
                        {
             y=rs.getInt(1);
          }
                        if(y==0)
                                System.out.println("this employee_id doesn't exist int kmit");
          else
                        {
           String query = "Select Name,department_id from Kmit_employee where
employee_id="'+n+"';";
                         stmt=conn.prepareStatement(query);
                         rs=stmt.executeQuery();
                         int temp_id=0;
                         while(rs.next())
                         {
                                 System.out.println("Name of the Employee : "+rs.getString(1));
                                 temp_id=rs.getInt(2);
                                 break;
                         }
          String query1="Select department_name,salary from Kmit_departments where
department_id="'+temp_id+"'";
                         stmt = conn.prepareStatement(query1);
          rs = stmt.executeQuery();
          while(rs.next())
                        {
            System.out.println("Name of the Department of Employee : "+rs.getString(1));
```

```
                    System.out.println("Department Id of Employee's Department : "+temp_id);
                    System.out.println("Salary of the Employee : "+rs.getInt(2));
                    break;
                }
            }
        }
        catch(SQLException se)
        {
            se.printStackTrace();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
*/
```

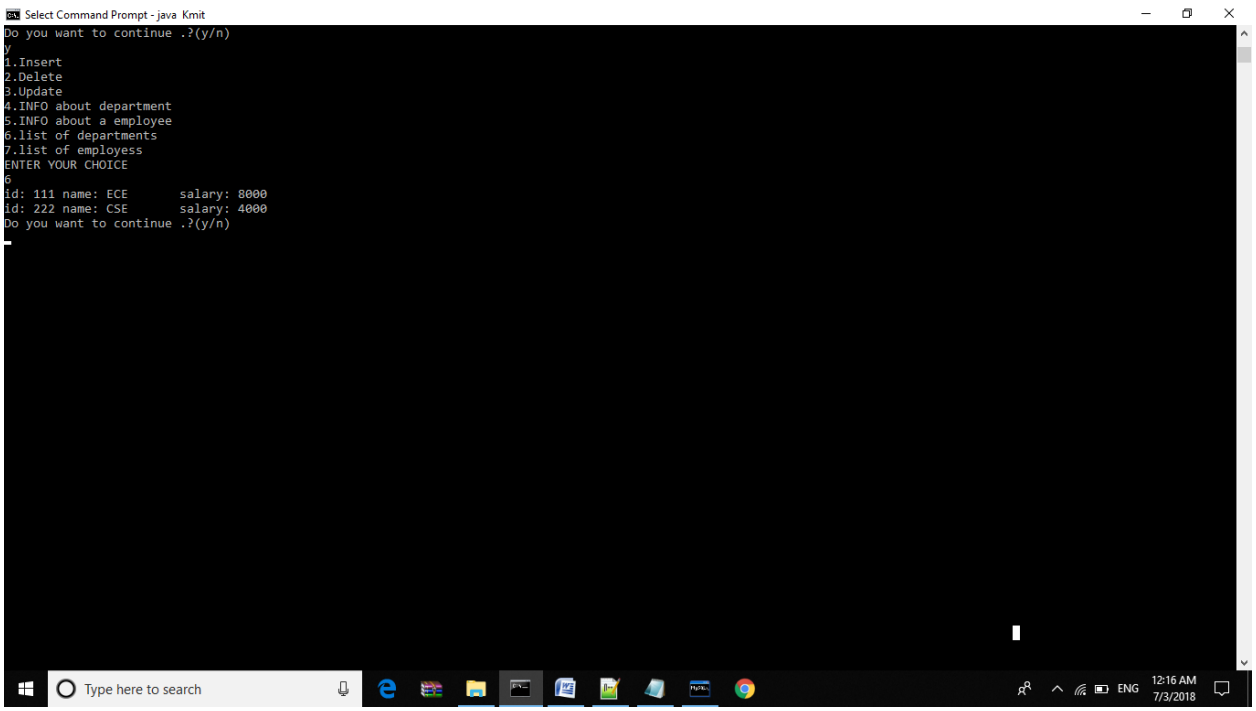


7. listOfDepartments

The listOfDepartments class provides the list of departments. It is called when the user enter 6 as the input. It has a method departmentsin kmit which when called dispalys the list of departments in kmit. It doesn't need any parameters.

Code of listOfDepartments Class

```
/*
class listOfDepartments extends Kmit
{
        public void departmentsInKmit()
        {
                try
                {
                        String sql="Select department_id,department_name,salary From Kmit.Kmit_departments;";
                        stmt=conn.prepareStatement(sql);
                        rs=stmt.executeQuery();

                   while(rs.next())
                   {
                        int id=rs.getInt("department_id");
                        String name=rs.getString("department_name");
                            int sal=rs.getInt("salary");
                        System.out.print("id: "+id+"\t");
                        System.out.print("name: "+name+"\t");
                        System.out.print("salary: "+sal+"\t");
                            System.out.println();
                   }
                }
                catch(SQLException se)
            {
                 //handling errors of jdbc
                 se.printStackTrace();
            }
       catch(Exception e)
            {
                 //handling errors of jdbc
                 e.printStackTrace();
            }
        }
*/
```


Select Command Prompt - java Kmit
Do you want to continue .?(y/n)
y
1.Insert
2.Delete
3.Update
4.INFO about department
5.INFO about a employee
6.list of departments
7.list of employess
ENTER YOUR CHOICE
6
id: 111 name: ECE salary: 8000
id: 222 name: CSE salary: 4000
Do you want to continue .?(y/n)
Type here to search
ENG
12:16 AM
7/3/2018

8. listOfEmployees

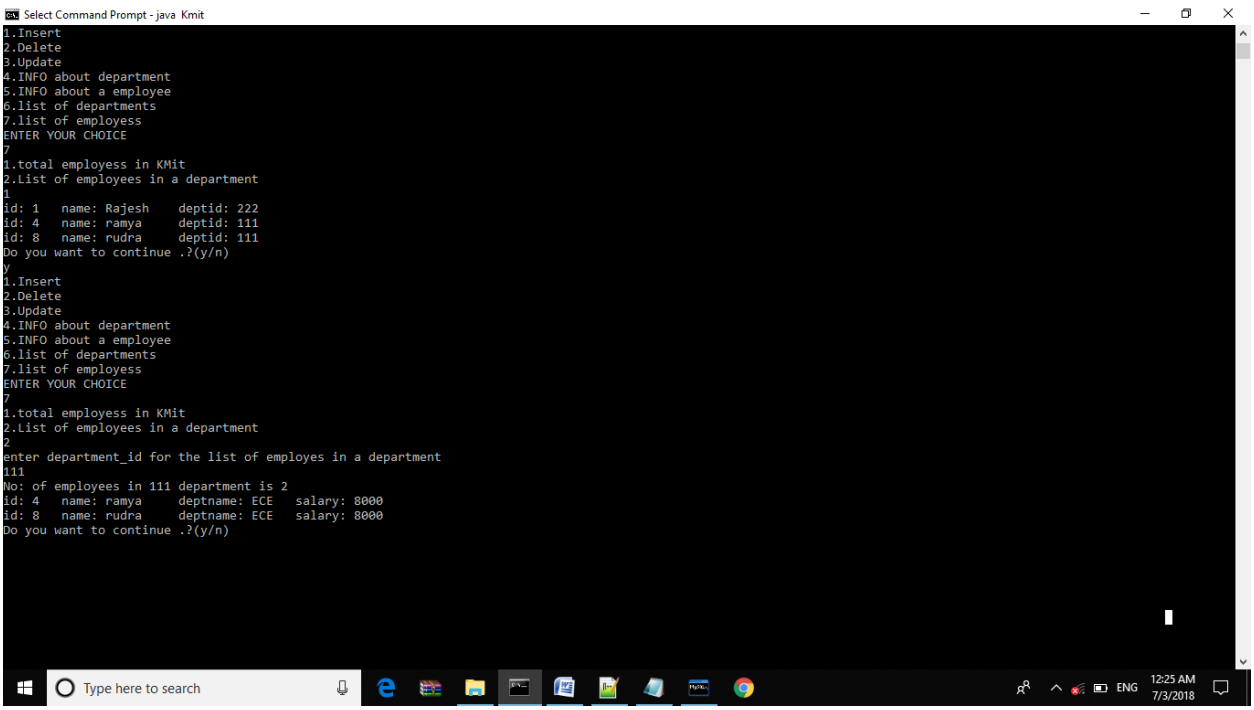
The listOfEmployees class is used to display total employees in kmit and employees in particular department. It is called when the user enter 7 as the input. It has two methods listofemplloyees and employeesInDepartment which displays list of all employees in kmit and list of employees in a particular department respectively when called by passing appropriate parameters. Two method are called in a switch case.

Code of listOfEmployees

```
/*
class listOfEmployees extends Kmit
{
    public void listofemplloyees()
    {

            int id=0;
            String name="";
            String depname="";
            int deptid=0;
            try{
                    String sql="Select * from kmit_employee";
                    stmt=conn.prepareStatement(sql);
                    rs=stmt.executeQuery();
               while(rs.next())
               {
                            id=rs.getInt("employee_id");
                            name=rs.getString("Name");
                            deptid=rs.getInt("department_id");
                            System.out.print("id: "+id+"\t");
                       System.out.print("name: "+name+"\t");
                       System.out.print("deptid: "+deptid+"\t");
                            System.out.println();

                    }
            }
            catch(SQLException se)
      {
              //handling errors of jdbc
              se.printStackTrace();
      }
     catch(Exception e)
      {
              //handling errors of jdbc
              e.printStackTrace();
      }
```

```
    }
    public void employeesInDepartment(int n)
    {
            InfoAboutDepartment iad1=new InfoAboutDepartment();
            if(iad1.getEmployees(n)==0)
            {
                    System.out.println("There are no Employees in this department");
            }
            else
            {
               int id=0;
               String name="";
               String depname="";
               int salary=0;
               int y=0;
              try
              {
                    String str1="SELECT count(*) FROM Kmit_departments WHERE
department_id ='"+n+"';";
                    stmt=conn.prepareStatement(str1);
                    rs=stmt.executeQuery();
                    while(rs.next())
                    {
          y=rs.getInt(1);
        }
                    if(y==0)
                            System.out.println("No department exists with this id");
                    else
                    {
                      String                                                    sql="select
Kmit_employee.employee_id,Kmit_employee.Name,Kmit_departments.department_name,K
mit_departments.salary          from          Kmit_employee,Kmit_departments          where
Kmit_employee.department_id='"+n+"' && Kmit_departments.department_id='"+n+"'; ";
                      rs=stmt.executeQuery(sql);
                while(rs.next())
                {
                            id=rs.getInt("employee_id");
                            name=rs.getString("Name");
                            depname=rs.getString("department_name");
                            salary=rs.getInt("salary");
                            System.out.print("id: "+id+"\t");
                      System.out.print("name: "+name+"\t");
                            System.out.print("deptname: "+depname+"\t");
                      System.out.print("salary: "+salary+"\t");
                            System.out.println();
```

```
                    }
                }
            }

            catch(SQLException se)
        {
            //handling errors of jdbc
            se.printStackTrace();
        }
      catch(Exception e)
        {
            //handling errors of jdbc
            e.printStackTrace();
        }

       }
    }
}
*/
```

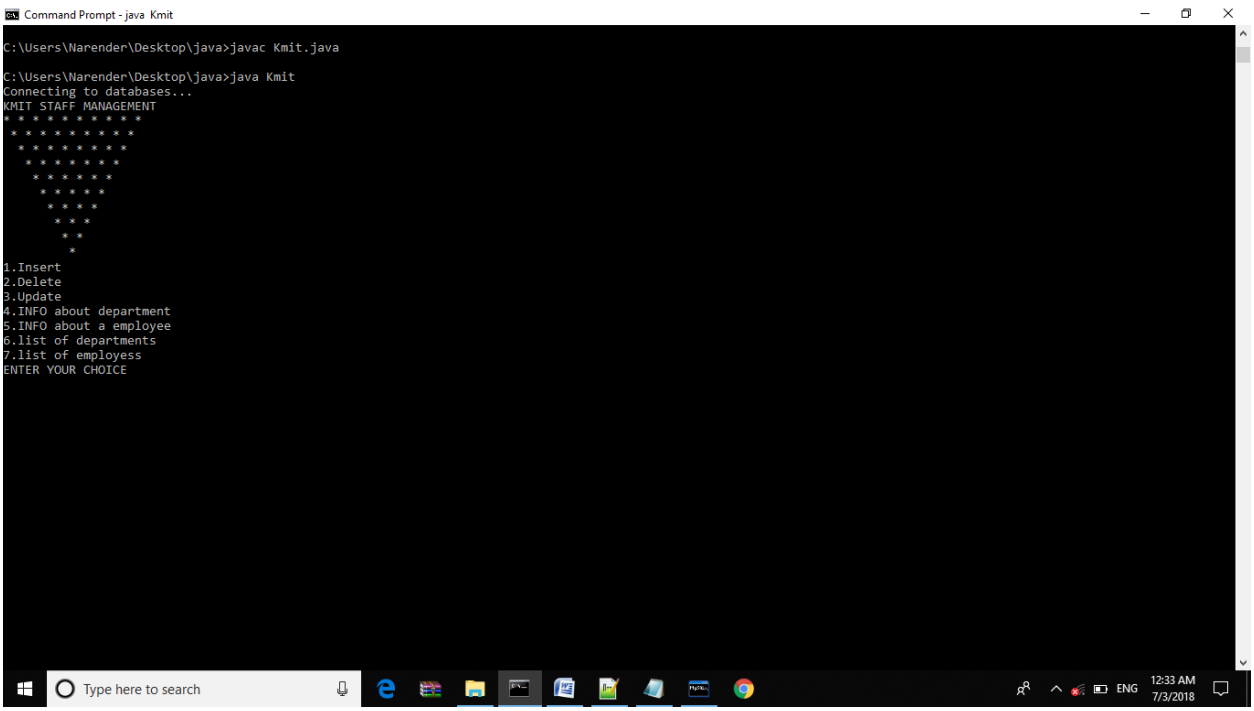


9. Logo

The logo class is used to print the downward pyramid with the title **STAFF MANGEMENT OF KMIT.** It doesn't have any methods but a default constructer of it contains the code of the pyramid. It is called outside the switch case

Code of Logo Class

```
/*
class Logo
{
   Logo()
     {
            System.out.println("KMIT STAFF MANAGEMENT");
            for(int i=0;i<10;i++)
            {
                  for(int j=0;j<i;j++)
                  {
                        System.out.print(" ");
                  }
                  for(int k=i;k>=i&&k<10;k++)
                  {
                        System.out.print("*");
                        System.out.print(" ");
                  }
            System.out.println();
            }

     }

}
*/
```


Command Prompt - java Kmit
C:\Users\Narender\Desktop\java>javac Kmit.java
C:\Users\Narender\Desktop\java>java Kmit
Connecting to databases...
KMIT STAFF MANAGEMENT
1.Insert
2.Delete
3.Update
4.INFO about department
5.INFO about a employee
6.list of departments
7.list of employess
ENTER YOUR CHOICE
Type here to search
ENG
12:33 AM
7/3/2018

The End