جامعة جدة
University of Jeddah

College of Science and Computer Engineering

Department of Science & AI

CCCS 224: Data Structure

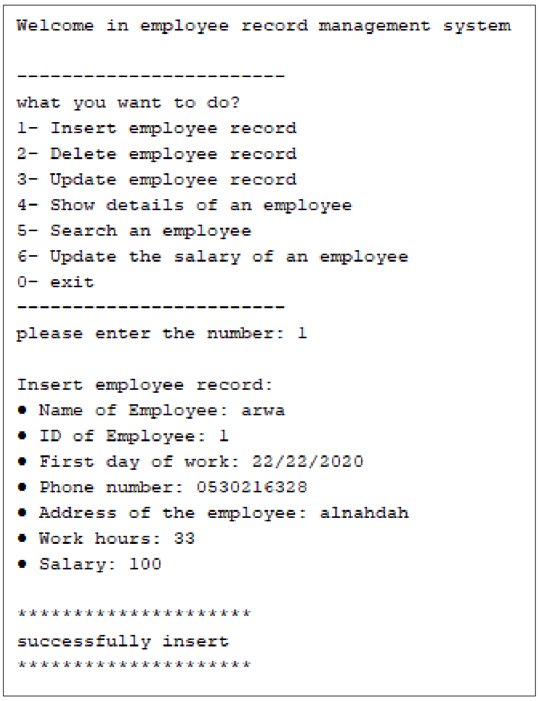
Data structures Project:
Employee record management system using linked list.

Name: **Arwa Omar Sait**

Fall 2023

Output:

a. Output of Insert employee record:



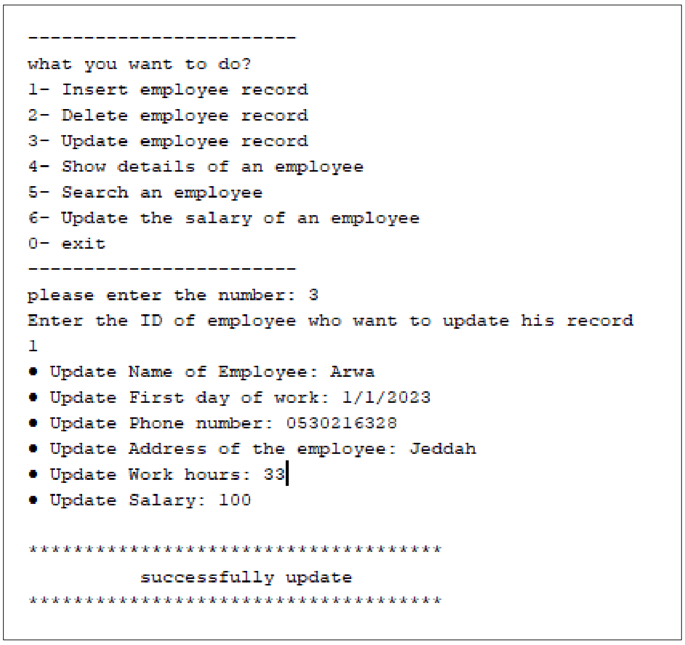
```
Welcome in employee record management system

------------------------
what you want to do?
1- Insert employee record
2- Delete employee record
3- Update employee record
4- Show details of an employee
5- Search an employee
6- Update the salary of an employee
0- exit
------------------------
please enter the number: 1

Insert employee record:
• Name of Employee: arwa
• ID of Employee: 1
• First day of work: 22/22/2020
• Phone number: 0530216328
• Address of the employee: alnahdah
• Work hours: 33
• Salary: 100

********************
successfully insert
********************
```

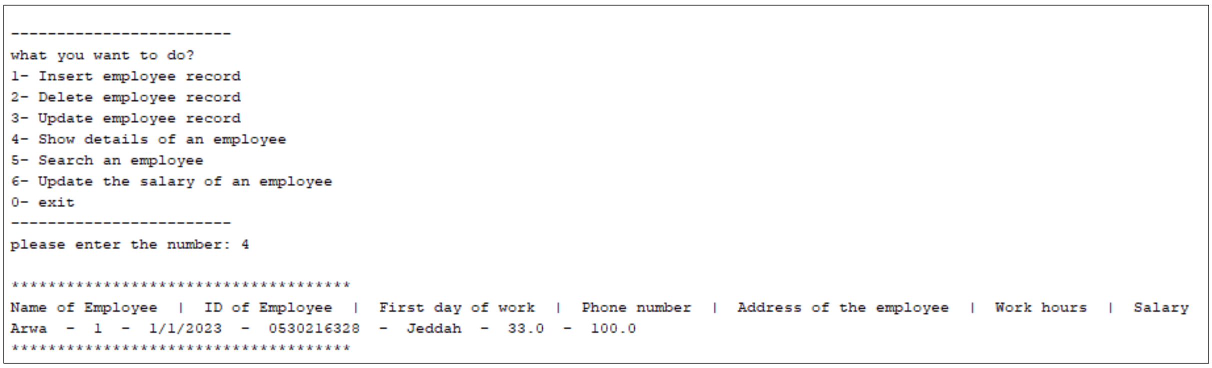
b. Update employee record



```
------------------------
what you want to do?
1- Insert employee record
2- Delete employee record
3- Update employee record
4- Show details of an employee
5- Search an employee
6- Update the salary of an employee
0- exit
------------------------
please enter the number: 3
Enter the ID of employee who want to update his record
1
• Update Name of Employee: Arwa
• Update First day of work: 1/1/2023
• Update Phone number: 0530216328
• Update Address of the employee: Jeddah
• Update Work hours: 33
• Update Salary: 100

************************************
          successfully update
************************************
```

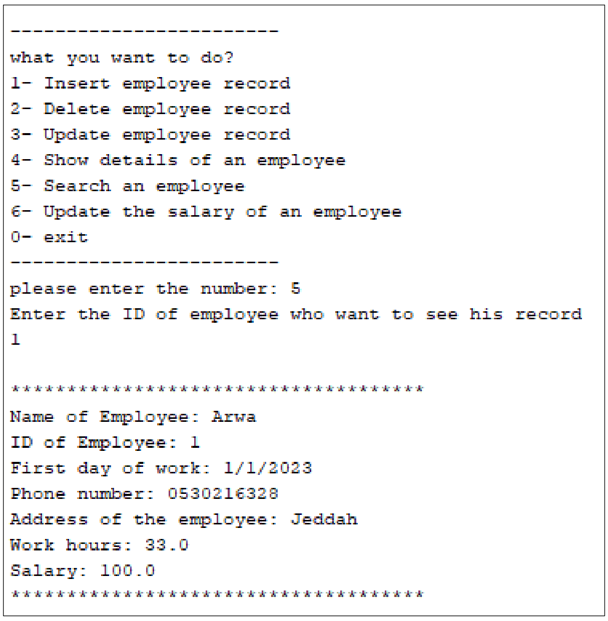
c. Show details of an employee



```
------------------------
what you want to do?
1- Insert employee record
2- Delete employee record
3- Update employee record
4- Show details of an employee
5- Search an employee
6- Update the salary of an employee
0- exit
------------------------
please enter the number: 4

*************************************
Name of Employee  |  ID of Employee  |  First day of work  |  Phone number  |  Address of the employee  |  Work hours  |  Salary
Arwa  -  1  -  1/1/2023  -  0530216328  -  Jeddah  -  33.0  -  100.0
*************************************
```

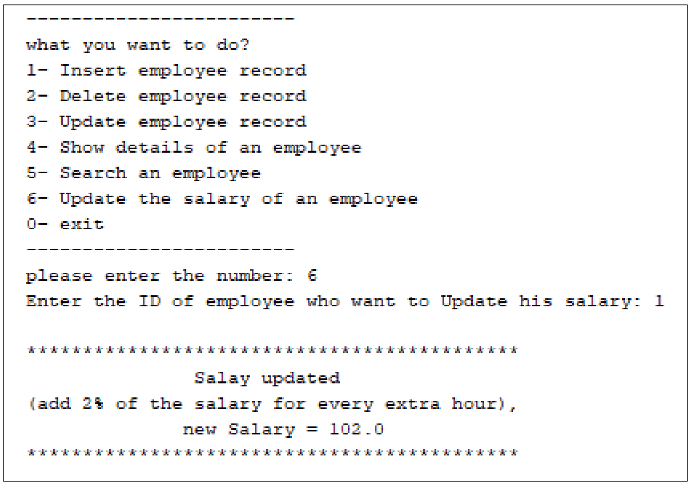
d. Search an employee



```
------------------------
what you want to do?
1- Insert employee record
2- Delete employee record
3- Update employee record
4- Show details of an employee
5- Search an employee
6- Update the salary of an employee
0- exit
------------------------
please enter the number: 5
Enter the ID of employee who want to see his record
1

*************************************
Name of Employee: Arwa
ID of Employee: 1
First day of work: 1/1/2023
Phone number: 0530216328
Address of the employee: Jeddah
Work hours: 33.0
Salary: 100.0
*************************************
```

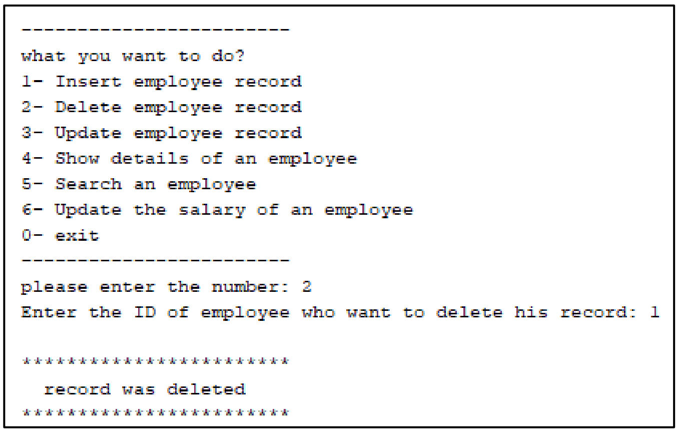
e. Update the salary of an employee



```
------------------------
what you want to do?
1- Insert employee record
2- Delete employee record
3- Update employee record
4- Show details of an employee
5- Search an employee
6- Update the salary of an employee
0- exit
------------------------
please enter the number: 6
Enter the ID of employee who want to Update his salary: 1

********************************************
               Salay updated
(add 2% of the salary for every extra hour),
              new Salary = 102.0
********************************************
```

f. Delete employee record



```
------------------------
what you want to do?
1- Insert employee record
2- Delete employee record
3- Update employee record
4- Show details of an employee
5- Search an employee
6- Update the salary of an employee
0- exit
------------------------
please enter the number: 2
Enter the ID of employee who want to delete his record: 1

************************
  record was deleted
************************
```

Code:

```
package employee_record_management;

/*
 * @author arwa omar sait
 * ID: 2111782
 * Section: EC1
 */

import java.util.InputMismatchException;
import java.util.Scanner;

public class Employee_Record_Management {

    ER_node head = null;

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        int chosen = 7;

        Employee_Record_Management linkList = new
Employee_Record_Management();
        System.out.println("Welcome in employee record management system \n");

        while (chosen != 0){
            System.out.println("------------------------");
            System.out.println("what you want to do?");
            System.out.println("1- Insert employee record\n" +
                              "2- Delete employee record\n" +
                              "3- Update employee record\n" +
                              "4- Show details of an employee\n" +
                              "5- Search an employee\n" +
                              "6- Update the salary of an employee\n" +
                              "0- exit");
            System.out.println("------------------------");
            System.out.print("please enter the number: ");

            try{
            chosen = input.nextInt();
            }catch(InputMismatchException e){
                System.out.println("\n*************************");
                System.out.println("    !!wrong input!! ");
                System.out.println("************************\n\n");
            }

            if(chosen == 1){
                linkList.Create_Record();
            }else if(chosen == 2){

                if(linkList.Delete_Record() == 0){
                    System.out.println("\n************************");
                    System.out.println("  record was deleted");
                    System.out.println("************************\n\n");
                }else{
                    System.out.println("\n*********************************");
                    System.out.println("there is not record has this id");
                    System.out.println("*********************************\n\n");
                }

            }else if(chosen == 3){
                linkList.Update_employee_record();
            }else if(chosen == 4){
                linkList.Show_Record();
            }else if(chosen == 5){
                linkList.Search_employee();
            }else if(chosen == 6){
                linkList.Update_salary();
            }else if(chosen == 0){
                System.exit(0);
            }else{
                System.out.println("\n************************");
                System.out.println("!!wrong input!! , Try again");
                System.out.println("************************\n\n");
            }
        }
    }

    //Insert employee record:
    public void Create_Record (){
        Scanner input = new Scanner(System.in);

        System.out.println("\nInsert employee record: ");

        System.out.print("• Name of Employee: ");
        String name = input.next();

        try{
```

```
System.out.print("• ID of Employee: ");
int ID = input.nextInt();

if(Check_Record(ID)){
   System.out.println("\n******************************************");
   System.out.println("        Record Already Exist!!");
   System.out.println("********************************************\n\n");
}else{
   System.out.print("• First day of work: ");
   String day = input.next();

   System.out.print("• Phone number: ");
   String Phone_number = input.next();

   System.out.print("• Address of the employee: ");
   String Address = input.next();

   try{
      System.out.print("• Work hours: ");
      double Work_hours = input.nextDouble();

      System.out.print("• Salary: ");
      double Salary = input.nextDouble();

      ER_node new_node = new ER_node(name, ID, day, Phone_number,
                              Address, Work_hours, Salary);

      if(head == null || head.ID >= new_node.ID){
         new_node.next = head;
         head = new_node;
      }else{
         ER_node temp = head;
         while (temp.next != null && temp.next.ID < new_node.ID)
            temp = temp.next;
         new_node.next = temp.next;
         temp.next = new_node;
      }


      System.out.println("\n************************");
      System.out.println(   "successfully insert");
      System.out.println(  "************************\n\n");

   }catch(InputMismatchException r){
      System.out.println("\n******************************************");
```

```
                System.out.println("          !!wrong input!! ");
                System.out.println(" Work hours & Salary should be double");

System.out.println("****************************************\n\n");
            }
        }
      }catch(InputMismatchException e){
            System.out.println("\n**************************************");
            System.out.println("          !!wrong input!!");
            System.out.println("        ID should be integer");
            System.out.println("****************************************\n\n");
      }
   }


   /*shows the record. similar to printing all elements of the Linked list.*/
   public void Show_Record(){
      if(head == null){
         System.out.println("\n**************************************");
         System.out.println(  "There is not record have this ID!");
         System.out.println("**************************************\n\n");
      }else{
         System.out.println("\n**************************************");
         System.out.println("Name of Employee  |" +
                         "  ID of Employee  |" +
                         "  First day of work  |" +
                         "  Phone number  |" +
                         "  Address of the employee  |" +
                         "  Work hours  |" +
                         "  Salary");
         ER_node temp = head;
         while(temp != null){
            System.out.println(temp);
            temp = temp.next;
         }
         System.out.println("**************************************\n\n");
     }
   }

   /*return true if the Record Already Exist else false
   uses concept of checking for a Node with given Data in a linked list.*/
   public boolean Check_Record(int id){
      ER_node temp = head;
      while(temp != null){
         if(id == temp.ID)
```

```
            return true;
        temp = temp.next;
    }
    return false;
}

/*Delete Record: integer returning
 *-1 if no such record with a given roll number is found
 *otherwise it deletes the node and returns 0.*/
public int Delete_Record(){
    Scanner input = new Scanner(System.in);
    System.out.print("Enter the ID of employee who want to delete his record: ");
    try{
        int id = input.nextInt();

        if(!Check_Record(id)){
        return -1;
        }else{

        ER_node temp = head, prev = null;

        if(temp != null && temp.ID == id){
        head = temp.next;
        return 0;
        }

        while (temp != null && temp.ID != id){
            prev = temp;
            temp = temp.next;
        }

        prev.next = temp.next;
        return 0;
        }
        }catch(InputMismatchException e){
            System.out.println("\n*************************************");
            System.out.println("          !!wrong input!!");
            System.out.println("        ID should be integer");
            System.out.println("***************************************\n\n");
        }
    return -1;
}

/*Update salary: add 2% of the salary for every extra hour.
By default, 32 hours are required for every employee.*/
```

```
  public void Update_salary(){

    Scanner input = new Scanner(System.in);
    System.out.print("Enter the ID of employee who want to Update his salary: ");
    try{
      int id = input.nextInt();

      if(!Check_Record(id)){
        System.out.println("\n**************************************");
        System.out.println(  "There is not record have this ID!");
        System.out.println("**************************************\n\n");
      }else{

        ER_node temp = head;

        while(temp != null && temp.ID != id)
          temp = temp.next;

        if(temp.Work_hours > 32 ){
          double add = 0.02 * temp.Salary;
          for(int i = 32; i<temp.Work_hours; i++)
            temp.Salary = temp.Salary + add;


System.out.println("\n*********************************************");
          System.out.println("              Salay updated\n"
                  + "(add 2% of the salary for every extra hour),"
                  + "\n             new Salary = " + temp.Salary);

System.out.println("********************************************\n\n");
        }else{

System.out.println("\n******************************************");
          System.out.println("        can not Update salary.\n"
                    +"Update only for extra hour (more than 32)");

System.out.println("******************************************\n\n");
        }
    }
    }catch(InputMismatchException e){
      System.out.println("\n**************************************");
      System.out.println("          !!wrong input!!");
      System.out.println("        ID should be integer");
      System.out.println("***************************************\n\n");
    }
    }

    public void Update_employee_record(){
        Scanner input = new Scanner(System.in);

        System.out.println("Enter the ID of employee who want to update his record");
        try{
            int ID_emp = input.nextInt();

            if(Check_Record(ID_emp)){

                ER_node temp = head;
                while(temp != null && temp.ID != ID_emp)
                    temp = temp.next;

                System.out.print("• Update Name of Employee: ");
                temp.Name = input.next();

                System.out.print("• Update First day of work: ");
                temp.First_day_of_work = input.next();

                System.out.print("• Update Phone number: ");
                temp.Phone_number = input.next();

                System.out.print("• Update Address of the employee: ");
                temp.Address = input.next();

                try{

                    System.out.print("• Update Work hours: ");
                    temp.Work_hours = input.nextDouble();

                    System.out.print("• Update Salary: ");
                    temp.Salary = input.nextDouble();

                    System.out.println("\n**************************************");
                    System.out.println("        successfully update");
                    System.out.println("**************************************\n\n");

                }catch(InputMismatchException e){
                    System.out.println("\n**************************************");
                    System.out.println("         !!wrong input!! ");
                    System.out.print(" Work hours & Salary should be double");

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

```
            }
        }else{
            System.out.println("\n**************************************");
            System.out.println(  "There is not record have this ID!");
            System.out.println("**************************************\n\n");
        }
    }catch(InputMismatchException e){
        System.out.println("\n**************************************");
        System.out.println("          !!wrong input!!");
        System.out.println("        ID should be integer");
        System.out.println("**************************************\n\n");
    }
}

public void Search_employee(){
    Scanner input = new Scanner(System.in);
    System.out.println("Enter the ID of employee who want to see his record");
    try{
        int id_emp = input.nextInt();

        if(head == null){
            System.out.println("\n**************************************");
            System.out.println( "  There is no record have this ID!");
            System.out.println("**************************************\n");
        }else{
            ER_node temp = head;
            while(temp != null && temp.ID != id_emp)
                temp = temp.next;

            System.out.println("\n**************************************");
            System.out.println("Name of Employee: " + temp.Name +
                               "\nID of Employee: " + temp.ID +
                               "\nFirst day of work: " + temp.First_day_of_work +
                               "\nPhone number: " + temp.Phone_number +
                               "\nAddress of the employee: " + temp.Address +
                               "\nWork hours: " + temp.Work_hours +
                               "\nSalary: " + temp.Salary );
            System.out.println("**************************************\n");
        }
    }catch(InputMismatchException e){
        System.out.println("\n**************************************");
        System.out.println("          !!wrong input!!");
        System.out.println("        ID should be integer");
        System.out.println("**************************************\n\n");
    }
    }
}





/**
 * @author arwao
 */
class ER_node {

    ER_node next;
    int ID;
    String Name, First_day_of_work, Phone_number, Address;
    double Work_hours, Salary;

    public ER_node(String name, int ID, String day,
    String Phone_number, String Address,double Work_hours,double Salary){
        Name = name;
        this.ID = ID;
        First_day_of_work = day;
        this.Phone_number = Phone_number;
        this.Address = Address;
        this.Work_hours = Work_hours;
        this.Salary = Salary;
        next = null;
    }

    @Override
    public String toString(){

        String employee_Record =  Name + "  -  " + ID + "  -  " + First_day_of_work +
                                "  -  " + Phone_number + "  -  " + Address +
                                "  -  " + Work_hours + "  -  " + Salary;
        return employee_Record;
    }
}
```