

JAVA ASSIGNMENT - 5

SESSION 2021-2022

LAB REPORT SUBMITTED

By:

Vivek Kumar Choudhary

(20204234)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD

PRAYAGRAJ, INDIA- 211004

1. You are required to create a database that has employee identification, employee name, fathers name, address, date of birth, designation, hourly payments rate and maximum payments that an employee can be paid in month for 100 hours. You may use the array concept for storing the records of 100 employees

Code:-

```
package com.company;

class Emp {
    private String name;
    private String fathersName;
    private String address;
    private String dob;
    private String designation;
    private double hoursPayement;
    private double maxPayement;

    public String getName() {
        return name;
    }

    public String getFathersName() {
        return fathersName;
    }

    public String getAddress() {
        return address;
    }

    public String getDob() {
        return dob;
    }

    public String getDesignation() {
        return designation;
    }

    public double getHoursPayement() {
        return hoursPayement;
    }

    public double getMaxPayement() {
        return maxPayement;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setFathersName(String fathersName) {
        this.fathersName = fathersName;
    }
}
```

```

    }

    public void setAddress(String address) {
        this.address = address;
    }

    public void setDob(String dob) {
        this.dob = dob;
    }

    public void setDesignation(String designation) {
        this.designation = designation;
    }

    public void setHoursPayment(double hoursPayment) {
        this.hoursPayment = hoursPayment;
    }

    public void setMaxPayment(double maxPayment) {
        this.maxPayment = maxPayment;
    }

    public Emp(String name, String fathersName, String address, String dob, String
designation, double hoursPayment, double maxPayment) {
        this.name = name;
        this.fathersName = fathersName;
        this.address = address;
        this.dob = dob;
        this.designation = designation;
        this.hoursPayment = hoursPayment;
        this.maxPayment = maxPayment;
    }
}

// MAIN METHOD
public class q1 {

    public static void main(String[] args) {

        Emp[] employe=new Emp[100];
        employe[0]=new Emp("Vivek","Prem
Narayan","Ujjain","07/02/2001","Student",20000,4000000);
        // write your code here
        System.out.println("Name: "+employe[0].getName());
        System.out.println("Fathers Name: "+employe[0].getFathersName());
        System.out.println("D.o.b: "+employe[0].getDob());
    }
}

```

```
System.out.println("Address: "+employee[0].getAddress());  
System.out.println("Designation: "+employee[0].getDesignation());  
System.out.println("Hours Payement: "+employee[0].getHoursPayement());  
System.out.println("Maximum Payement: "+employee[0].getMaxPayement());  
}  
}
```

Output:-

```
Name: Vivek  
Fathers Name: Prem Narayan  
D.o.b: 07/02/2001  
Address: Ujjain  
Designation: Student  
Hours Payement: 20000.0  
Maximum Payement: 4000000.0
```

2. You are required to write classes for student details (name, registration number, fathers name), address (current address, permanent address, contact number, email Id) using inheritance concept where address is child class.

Code:-

```
package com.company;  
  
class details{  
  
    String name;  
  
    int reg_no;  
    String father_name;
```

```

public details(String name, int reg_no, String father_name) {
    this.name = name;
    this.reg_no = reg_no;
    this.father_name = father_name;
}

}

class address extends details{

    String current_adrs;

    String permanent_adrs;

    String contact_no;

    String email;


    public address(String name, int reg_no, String father_name, String current_adrs, String
    permanent_adrs, String contact_no, String email) {

        super(name, reg_no, father_name);

        this.current_adrs = current_adrs;

        this.permanent_adrs = permanent_adrs;

        this.contact_no = contact_no;

        this.email = email;

    }

}

public class q2 {

    public static void main(String[] args) {

        address student = new address("Vivek", 20204234, "PremNarayan", "MNNIT
        Allahabad", "Ujjain", "8770976496", "vivekchoudhary19915@gmail.com");

        System.out.println(student.name);
    }
}

```

```
System.out.println(student.reg_no);

System.out.println(student.father_name);

System.out.println(student.current_adrs);

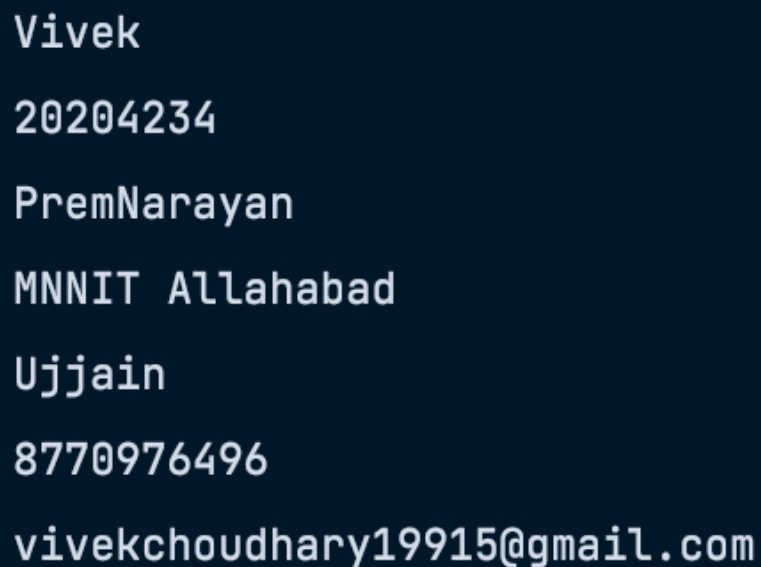
System.out.println(student.permanent_adrs);

System.out.println(student.contact_no);

System.out.println(student.email);
}

}
```

Output:-

A screenshot of a program's output displayed on a dark blue background with white text. The output consists of seven lines, each representing a different attribute of a student object. The attributes and their values are: Name (Vivek), Registration Number (20204234), Father's Name (PremNarayan), Current Address (MNNIT Allahabad), Permanent Address (Ujjain), Contact Number (8770976496), and Email (vivekchoudhary19915@gmail.com).

```
Vivek
20204234
PremNarayan
MNNIT Allahabad
Ujjain
8770976496
vivekchoudhary19915@gmail.com
```

3. Further you are required to extend the problem no 2 for addition of the classes, using inherit concept, Courses Registered (Course name, credit, course number, associated teachers), Academic records(name of degree, year of passing, university/board, percentage marks), Payments Details (bank name, transaction id, amount, date).

Code:-

```
package com.company;

class detail{

    String name;

    int reg_no;

    String father_name;


    public detail(String name, int reg_no, String father_name) {
        this.name = name;

        this.reg_no = reg_no;

        this.father_name = father_name;
    }
}

class address2 extends detail{

    String current_adrs;

    String permanent_adrs;

    String contact_no;

    String email;


    public address2(String name, int reg_no, String father_name, String current_adrs, String
    permanent_adrs, String contact_no, String email) {

        super(name, reg_no, father_name);

        this.current_adrs = current_adrs;

        this.permanent_adrs = permanent_adrs;
        this.contact_no = contact_no;
        this.email = email;
    }
}
```

```
}
```

```
class courses extends address2{
```

```
    String course_name;
```

```
    double credit;
```

```
    int course_no;
```

```
    int associate_teachers;
```

```
    public courses(String name, int reg_no, String father_name, String current_adrs, String permanent_adrs, String contact_no, String email, String course_name, double credit, int course_no, int associate_teachers) {
```

```
        super(name, reg_no, father_name, current_adrs, permanent_adrs, contact_no, email);
```

```
        this.course_name = course_name;
```

```
        this.credit = credit;
```

```
        this.course_no = course_no;
```

```
        this.associate_teachers = associate_teachers;
```

```
    }
```

```
}
```

```
class academic_record extends courses{
```

```
    String name_of_degree;
```

```
    int year_of_passing;
```

```
    String university;
```

```
    double percentage;
```

```
    public academic_record(String name, int reg_no, String father_name, String current_adrs, String permanent_adrs, String contact_no, String email, String
```



```
course_name, double credit, int course_no, int associate_teachers, String  
name_of_degree, int year_of_passing, String university, double percentage) {
```

```
    super(name, reg_no, father_name, current_adrs, permanent_adrs, contact_no, email,  
course_name, credit, course_no, associate_teachers);
```

```
    this.name_of_degree = name_of_degree;
```

```
    this.year_of_passing = year_of_passing;
```

```
    this.university = university;  
    this.percentage = percentage;
```

```
}
```

```
}
```

```
class payments extends academic_record{
```

```
    String Bank_name;
```

```
    String transection_id;
```

```
    double amount;
```

```
    String date;
```

```
    public payments(String name, int reg_no, String father_name, String current_adrs,  
String permanent_adrs, String contact_no, String email, String course_name, double  
credit, int course_no, int associate_teachers, String name_of_degree, int year_of_passing,  
String university, double percentage, String bank_name, String transection_id, double  
amount, String date) {
```

```
        super(name, reg_no, father_name, current_adrs, permanent_adrs, contact_no, email,  
course_name, credit, course_no, associate_teachers, name_of_degree, year_of_passing,  
university, percentage);
```

```
        Bank_name = bank_name;
```

```
        this.transection_id = transection_id;
```

```
        this.amount = amount;
```

```
        this.date = date;
```

```

    }
}

public class q3 {

    public static void main(String[] args) {

        payments student = new payments("Vivek",20204234,"Prem
Narayan","Ujjain","MP","9516046427","vivekchoudhary19915@gmail.com","oops
Lab",3,12301,3,"Engineering",2024,"MNNIT",91.5,"SBI
Bank","PK070533",10520,"10/12/2021");

        System.out.println(student.name);

        System.out.println(student.reg_no);

        System.out.println(student.father_name);

        System.out.println(student.current_adrs);

        System.out.println(student.permanent_adrs);

        System.out.println(student.contact_no);

        System.out.println(student.course_name);

        System.out.println(student.credit);

        System.out.println(student.course_no);

        System.out.println(student.associate_teachers);

        System.out.println(student.name_of_degree);

        System.out.println(student.year_of_passing);

        System.out.println(student.university);

        System.out.println(student.percentage);

        System.out.println(student.Bank_name);

        System.out.println(student.transection_id);

        System.out.println(student.amount);
    }
}

```

```
System.out.println(student.date);  
}  
  
}
```

OUTPUT:-

```
Vivek  
20204234  
Prem Narayan  
Ujjain  
MP  
9516046427  
oops Lab  
3.0  
12301  
3  
Engineering  
2024  
MNNIT  
91.5  
SBI Bank  
PK070533  
10520.0  
10/12/2021
```

4. You are required to create array of string of 10 elements. Further sort the array in ascending order. The string comparison can be with or without case sensitive.

CODE:-

```
package com.company;

import java.util.Arrays;
public class q4
{
    public static void main(String args[]) {
        String[] friends = {"shailendra", "ritik", "vikram","somu","vivek", "tarun",
"sandeep", "Subodh", "saurabh", "ankit", "prince"};
        int size = friends.length;
        System.out.println("Unsorted list is ");
        System.out.println();
        System.out.println(Arrays.toString(friends));
        for(int i = 0; i<size-1; i++) {
            for (int j = i+1; j<friends.length; j++) {
                if(friends[i].compareTo(friends[j])>0) {
                    String temp = friends[i];
                    friends[i] = friends[j]; friends[j] = temp;
                }
            }
        }
        System.out.println("sorted list is ");

        System.out.println();
        System.out.println(Arrays.toString(friends));
    }
}
```

OUTPUT:-

```
Unsorted list is

[shailendra, ritik, vikram, somu, vivek, tarun, sandeep, Subodh, saurabh, ankit, prince]
sorted list is

[Subodh, ankit, prince, ritik, sandeep, saurabh, shailendra, somu, tarun, vikram, vivek]
```

5. You are given a paragraph consisting of words, Convert this para graph to tokens (word) to be stored in array of string in sorted order where sorted array does not have duplicate words.

CODE:-

```
package com.company;

import java.util.*;
public class q5 {
    //delete duplicate tokens
    static String[] deleteduplicate(String str[]) {
        int i,j=0;
        int l=str.length;
        String arr[]=new String[l]; arr[j++]=str[0]; for(i=1;i<l;i++)
        {
            if(str[i-1].compareTo(str[i])!=0) arr[j++]=str[i];
        }
        return arr; }
    //function to sort string array static
    String[] sort(String str[]) {
        String temp;
        int l=str.length;
        int i,j;
        for(i=0;i<l-1;i++) {
            for(j=0;j<l-i-1;j++) {
                if(str[j].compareTo(str[j+1])>0) {
                    temp=str[j];

                    str[j]=str[j+1];
                    str[j+1]=temp; }
            } }
        return str; }
    //main function
    public static void main(String args[]) {
        Scanner c =new Scanner(System.in); String para;
        int i;
        System.out.println("Enter a paragraph!!"); para=c.nextLine();
        //Passing the entered paragraph to string tokenizer class to break it into tokens when
        //space or , or . or ? *encountered
        StringTokenizer obj=new StringTokenizer(para,".,?/: ");
        //stores the number of tokens present
```

```

int ct=obj.countTokens();
String str[]=new String[ct];
i=0;

//storing these tokens in a string array while(obj.hasMoreTokens())
{
    str[i++]=obj.nextTokent(); }
//displaying the tokens
System.out.println("Paragraph after being broken into tokens::"); for(i=0;i<ct;i++)
{
    System.out.print(str[i]+" "); }
System.out.println();
//sorting the string array str=sort(str); System.out.print("After sorting\n");
//displaying after sorting for(i=0;i<ct;i++)
{
    System.out.print(str[i]+" "); }

//deleting the duplicate tokens
System.out.println();
System.out.println("After deleting the duplicate elements:");
str=deleteduplicate(str);
//displaying after deleting the duplicated tokens:: for(i=0;i<str.length;i++)
{
    System.out.print(str[i]+" "); }
} }

```

OUTPUT:-

```
Enter a paragraph!!
HELLO I AM VIVEK FROM MNNIT CSE
Paragraph after being broken into tokens::
HELLO I AM VIVEK FROM MNNIT CSE
After sorting
HELLO I AM VIVEK FROM MNNIT CSE
After deleting the duplicate elements:
HELLO I AM VIVEK FROM MNNIT CSE
Process finished with exit code 0
```

6. You are given a paragraph consisting continuous alphabets, Convert this para graph to tokens of fixed size. Store the tokens in array of string in sorted order where sorted array does not have duplicate words.

```
import java.util.*;
class ques_6
{
    public static void main(String[] args)
    {
        int size;
        String line, temp;

        Scanner s = new Scanner(System.in);

        System.out.println("Enter the paragraph to tokenize :");
```

```
line = s.nextLine();
```

```
System.out.println("Enter the size to tokenize :");
```

```
size = s.nextInt();
```

```
int token_count = line.length()/size + 1;
```

```
String[] arr = new String [token_count];
```

```
//Tokenizing the paragraph
```

```
int t = 0;
```

```
while(line.length() > size) {
```

```
    arr[t++] = line.substring(0, size);
```

```
    line = line.substring(size);
```

```
}
```

```
arr[t] = line;
```

```
//Sorting the strings
```

```
for (int i = 0; i < token_count; i++)
```

```
{
```

```
    for (int j = i + 1; j < token_count; j++) {
```

```
        if (arr[i].compareTo(arr[j])>0)
```

```
        {
```

```
            temp = arr[i];
```

```
            arr[i] = arr[j];
```

```
            arr[j] = temp;
```

```
        }
```



```
    }  
}
```

//Deleting the duplicates

```
int i = 0, dupli_count = 0;
```

```
while (i < token_count)
```

```
{
```

```
    int j = i+1;
```

```
    while (j < token_count && arr[i].compareTo(arr[j]) == 0)
```

```
    {
```

```
        arr[j] = "";
```

```
        dupli_count += 1;
```

```
        j++;
```

```
    }
```

```
    i = j;
```

```
}
```

```
String[] finalarr = new String [token_count-dupli_count];
```

```
for(int j = 0, k = 0; j < token_count; j++)
```

```
{
```

```
    if(arr[j].equals(""))
```

```
        continue;
```

```
    finalarr[k] = arr[j];
```

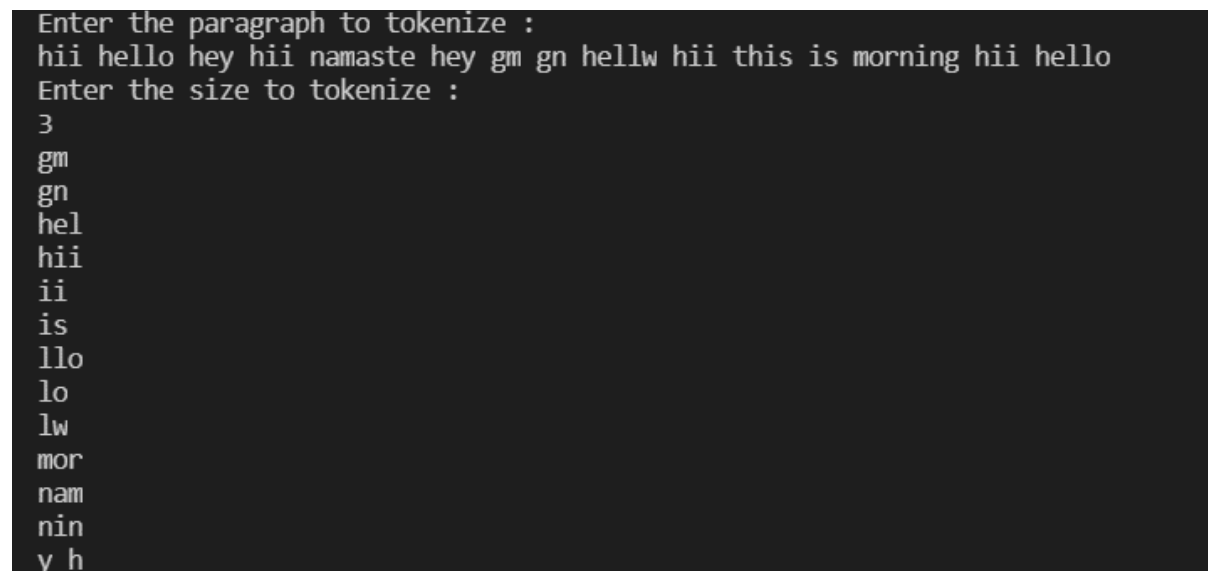
```
    k++;
```

```
}
```

```

        System.out.println("The tokens after sorting and removing the duplicates
are: ");
        for (int j = 0; j < token_count-dupli_count; j++)
        {
            System.out.println(finalarr[j]);
        }
    }
}

```



```

Enter the paragraph to tokenize :
hii hello hey hii namaste hey gm gn hellw hii this is morning hii hello
Enter the size to tokenize :
3
gm
gn
hel
hii
ii
is
llo
lo
lw
mor
nam
nin
y h

```

7. You are given a paragraph consisting continuous alphabets, Convert this para graph to tokens based on occurrence of particular characters. Store the tokens in array of string in sorted order where sorted array does not have duplicate words.

```

import java.util.Scanner;
import java.util.StringTokenizer;

```

```
public class ques_7 {

    public static void main(String[] args)
    {
        String line, alpha, temp;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the paragraph to tokenize :");
        line = s.nextLine();
        System.out.println("Enter the character to tokenize :");
        alpha = s.nextLine();
        //Tokenizing the paragraph
        StringTokenizer tokenz = new StringTokenizer(line, alpha);
        int token_count = tokenz.countTokens();
        String[] arr = new String [token_count];
        int t = 0;
        while (tokenz.hasMoreTokens())
        {
            arr[t++] = tokenz.nextToken();
        }
        //Sorting the strings
        for (int i = 0; i < token_count; i++)
        {
            for (int j = i + 1; j < token_count; j++) {
                if (arr[i].compareTo(arr[j])>0)
                {
                    temp = arr[i];
```

```

        arr[i] = arr[j];
        arr[j] = temp;
    }
}

//Deleting the duplicates
int i = 0, count_dupli = 0;
while (i < token_count)
{
    int j = i+1;
    while (j < token_count && arr[i].compareTo(arr[j]) == 0)
    {
        arr[j] = "";
        count_dupli += 1;
        j++;
    }
    i = j;
}

```

```

String[] newarr = new String [token_count-count_dupli];
for(int j = 0, k = 0; j < token_count; j++)
{
    if(arr[j].equals(""))
        continue;

    newarr[k] = arr[j];
}

```

```
        k++;  
    }
```

```
    System.out.println("The tokens after sorting and removing the  
duplicates are: ");
```

```
    for (int j = 0; j < token_count-count_dupli; j++)  
    {  
        System.out.println(newarr[j]);  
    }  
}  
}
```

```
Enter the paragraph to tokenize :  
hii friend welcome you all hii hello hey nothing dear hello hii friend  
Enter the character to tokenize :  
hii  
The tokens after sorting and removing the duplicates are:  
  
fr  
ello  
end  
end welcome you all  
ey not  
ng dear
```