Date created: Sep 23, 2021

Date modified: Sep 24, 2021

**INTRODUCTION**

User Registry is a program designed to handle employee registration data using variables and allow related employee operations such as transferring or promoting employees. The is a program designed to solve the following problem statement.

**Design a class called Employee that will characterize five attributes of a university employee such as date of birth, name, department, Basic salary, commuter allowance etc. Your program should have methods to calculate employee monthly salary and display the result as well as a method to re-designate an employee to a new position or even transfer the employee to another department**

In conclusion, the solution should allow users to easily register employees, calculate employee allowances and to promote or transfer employees.

**PROCEDURE**

A set of steps has been developed to deal with data related to user registry, an array data structure is also used to allow related operations. The program will follow the following procedures to solve the problem statement and create a user registry using Java

1. Create a blueprint for an object of type Employee
2. Add properties and fields to class Employee
3. Create a friendly interface in the main method to allow registry processes

In conclusion, the procedure followed should be efficient, have minimal memory usage associated with procedures and should be non-repetitive.

**IMPLEMENTATION**

import java.util.Scanner;

class Student {

//main method

public static void main(String[] args){

String stopAddCondition = "";

continueAdd()

stopAdd()

(stopAddCondition = addUser() ? continueAdd : stopAdd)){

};

}

//a function to add users to the database

public static String addUser(){

// myScanner object of class Scanner

Scanner myScanner = new Scanner(System.in);

//local variables

String stop\_condition = "";

String stud\_name = "";

int stud\_age = 0;

//ask for user input

System.out.println("Enter name and age");

stud\_name = myScanner.nextLine();

stud\_age = myScanner.nextInt();

// new\_student object of class NewStudent using user input as parameters

NewStudent new\_student = new NewStudent(stud\_name, stud\_age);

System.out.println("-----------Created new student-------------");

String gotten\_description = "";

gotten\_description = new\_student.CreateDescription();

new\_student.DescribeStudent(gotten\_description);

System.out.println("------------------Finished describing new student---------");

System.out.println("Finished? (Y/n)");

stop\_condition = myScanner.nextLine();

return stop\_condition;

}

}

//A blueprint for objects of type NewStudent

class NewStudent {

//local variables in NewStudent class

public String student\_name = "";

public int student\_age = 0;

//constructor for NewStudent

NewStudent(String namePassed, int agePassed){

this.student\_name = namePassed;

this.student\_age = agePassed;

}

//fetch name and age for NewStudent and return it

public String CreateDescription() {

String my\_description = "";

my\_description = "name: " + this.student\_name + ", " + "age: " + this.student\_age;

return( my\_description );

}

//display name and age for NewStudent

public void DescribeStudent(String gotten\_values){

System.out.println(gotten\_values);

}

}