DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

CU CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

Experiment: 1

Student Name: AKASH DEEP	**UID:22BCS10195**
Branch: CSE	**Section/Group: DL-902**
Semester: 6th	**Date of Performance:08/01/25**
Subject Name: Project Based Learning In Java with Lab	**Subject Code: 22CSH-359**

1. Aim: Create an application to save the employee information using arrays.

2. Objective: Given the following table containing information about employees of an organization, develop a small java application, which accepts employee id from the command prompt and displays the output.

3. Steps to Solve:

1. Input and Output:

- Input: Employee ID from the user.
- Output: Employee details or “No employee with empid: {empId}”.

2. Data Initialization: Store employee and designation details in 2D arrays.
3. Prompt for Input: Read employee ID using Scanner.
4. Search for Employee: Loop through employee data to find a matching ID.
5. Extract and Calculate:

- Use switch-case for designation and DA.
- Calculate salary as Salary = Basic + HRA + DA - IT.

6. Display Output: Print details if found; otherwise, display no-match message.
7. Close Scanner: Close the Scanner to release resources.

4. Code:

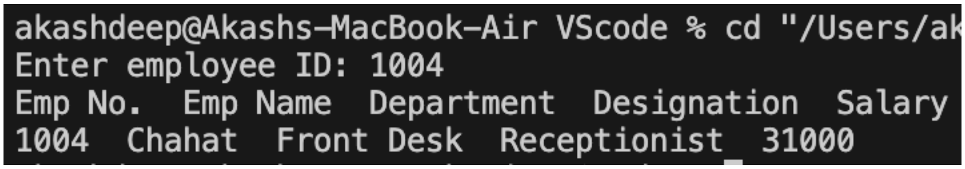
```
import java.util.*;


public class employee{
    public static void main (String args[]){
        String[][] employees = {
            {"1001", "Ashish", "01/04/2009", "e", "R&D", "20000", "8000", "3000"},
            {"1002", "Sushma", "23/08/2012", "c", "PM", "30000", "12000", "9000"},
            {"1003", "Rahul", "12/11/2008", "k", "Acct", "10000", "8000", "1000"},
            {"1004", "Chahat", "29/01/2013", "r", "Front Desk", "12000", "6000", "2000"},
            {"1005", "Ranjan", "16/07/2005", "m", "Engg", "50000", "20000", "20000"},
            {"1006", "Suman", "01/01/2000", "e", "Manufacturing", "23000", "9000", "4400"},
            {"1007", "Tanmay", "12/06/2006", "c", "PM", "29000", "12000", "10000"}
        };
        String[][] designations = {
            {"e", "Engineer", "20000"},
            {"c", "Consultant", "32000"},
            {"k", "Clerk", "12000"},
            {"r", "Receptionist", "15000"},
            {"m", "Manager", "40000"}
        };
        Scanner obj = new Scanner(System.in);
        System.out.print("Enter employee ID: ");
        String empId = obj.nextLine();
        boolean found=false;
```

```
for(String[] emp:employees){
    if(emp[0].equals(empId)){
        found=true;
        String desigCode = emp[3];
        String designation = "";
        int da = 0;
        switch (desigCode) {
            case "e":
                designation = "Engineer";
                da = 20000;
                break;
            case "c":
                designation = "Consultant";
                da = 32000;
                break;
            case "k":
                designation = "Clerk";
                da = 12000;
                break;
            case "r":
                designation = "Receptionist";
                da = 15000;
                break;
            case "m":
                designation = "Manager";
```

```
                        da = 40000;
                        break;
                }
                int basic = Integer.parseInt(emp[5]);
                int hra = Integer.parseInt(emp[6]);
                int it = Integer.parseInt(emp[7]);
                int salary = basic+hra+da-it;
                System.out.println("Emp No.  Emp Name  Department  Designation  Salary");
                System.out.printf("%s  %s  %s  %s  %d%n", emp[0], emp[1], emp[4], designation, salary);
                break;
            }
        }
        if(!found){
            System.out.println("No employee with empid:"+empId);
        }
        obj.close();
    }
}
```

5. Output:



```
akashdeep@Akashs-MacBook-Air VScode % cd "/Users/ak
Enter employee ID: 1004
Emp No.  Emp Name  Department  Designation  Salary
1004  Chahat  Front Desk  Receptionist  31000
```

6. Learning Outcome :

- Efficient use of 2D arrays for data storage and retrieval.
- Implementation of switch-case for logical decision-making.
- Dynamic user input handling and validation.
- Salary computation using arithmetic operations.
- Iterative search logic for record matching.