**Class & Objects**

1. Write a class named MyString which has an attribute “data” of type String and which performs the following operation
   1. Find the total length of the String
   2. Append another string with the value present in the attribute “data”.
   3. Count the number of vowels present in the String
   4. Check whether a given character is present in the value of the attribute “data”
   5. Display each character of the String
   6. Check whether the given String is equal to the value present in the attribute “data”.
   7. Display the characters of the data from 3rd position.
2. Write a class Employee with attributes empIde,empName, deptName(tech or admin),designation ( TM,PL,TL , Admin),salary
   1. calculate allowance based on designation

|  |  |
| --- | --- |
| **Design** | **Allowance** |
| PL | 1600 |
| TM | 1300 |
| TL | 1270 |
| Admin | 1500 |
| Others | 1100 |

* 1. calculate Tax (calculate monthly tax and deduct from the salary)

|  |  |
| --- | --- |
| >20,000 | = 13 % PA |
| 15000 - 20000 | = 12.5 % PA |
| 12500 – 14999 | = 11.25% PA |
| 7500-12499 | = 9.6 % PA |
| <7500 | = NIL |

* 1. calculate gross salary
  2. calculate net salary
  3. Create a Main class
     1. add employee (empId,emName,deptName,designation and salary is mandate)
     2. display employee detail with basic,gross and net of any employee
     3. generate toString method to display employee data on direct object name

1. Create an Employee class with empId, empName, empAddress, empEmail, empDesignation.
   1. Write a method to add,view, modify (except empid and empname) the employee details
   2. Write a method to add an employee without email
   3. The employee can be added without any designation. If not given “Trainee” should be the designation.

**Arrays and static**

1. Modify the above class employee with an extra attribute certifications done by employee (should be an array).
   1. Create appropriate getters and setters.
   2. Create a class called EmployeeUtility which has an employee array capable of holding 15 employees.
   3. The utility class should be able to add an employee to the array and View all employees. And count the number of employee objects created and display it.
2. Write a Program to implement Dictionary using array.
   1. Have the array size as 5.
   2. Dictionary class has 2 attributes name and meaning.
   3. A single word can have multiple meaning.
   4. Write a class called DictionaryUtility which has the method to add word and meaning to the array, method to display the meaning for a given word.
3. Modify the Employee and EmployeeUtility classes
   1. Modify the Employee class
      1. By adding a method which confirms that no 2 Employees have the same id. Use the equals method to do the check.
      2. Override the toString method to print the employee details.
   2. Modify the EmployeeUtility class to
      1. Define a sort method which sorts the employees in the array.
      2. Search an employee by their name and return the employee object, and by their id and return the Employee Object
   3. Test the newly added methods.