

Collection Questions

Colour Code:

Green – Print by the application

This is printed by the application

Blue – Sample Input given by user

This is printed by the application

Preparation

Create the Employee class as below.

```
class Employee
{
    int id, age;
    string name;
    double salary;

    public Employee()
    {
    }

    public Employee(int id, int age, string name, double salary)
    {
        this.id = id;
        this.age = age;
        this.name = name;
        this.salary = salary;
    }

    public void TakeEmployeeDetailsFromUser()
    {
        Console.WriteLine("Please enter the employee ID");
        id = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Please enter the employee name");
        name = Console.ReadLine();
        Console.WriteLine("Please enter the employee age");
        age = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Please enter the employee salary");
        salary = Convert.ToDouble(Console.ReadLine());
    }

    public override string ToString()
    {
        return "Employee ID : " + id + "\nName : " + name + "\nAge : " + age +
"\nSalary : " + salary;
    }

    public int Id { get => id; set => id = value; }
    public int Age { get => age; set => age = value; }
    public string Name { get => name; set => name = value; }
    public double Salary { get => salary; set => salary = value; }
}
```

Easy:

- 1) Create a C# console application which has a class with name "EmployeePromotion" that will take employee names in the order in which they are eligible for promotion.

- a. Example Input:

```
Please enter the employee names in the order of their eligibility for
promotion(Please enter blank to stop)
Ramu
Bimu
Somu
Gomu
Vimu
```

- b. Create a collection that will hold the employee names in the same order that they are inserted.
 - c. Hint – choose the correct collection that will preserve the input order (List)
- 2) Use the application created for question 1 and in the same class do the following
 - a. Given an employee name find his position in the promotion list
 - b. Example Input:

```
Please enter the employee names in the order of their eligibility for promotion
Ramu
Bimu
Somu
Gomu
Vimu
Please enter the name of the employee to check promotion position
Somu
"Somu" is the the position 3 for promotion.
```

- c. Hint – Choose the correct method that will give back the index (IndexOf)
- 3) Use the application created for question 1 and in the same class do the following
 - a. The application seems to be using some excess memory for storing the name, contain the space by using only the quantity of memory that is required.
 - b. Example Input:

```
Please enter the employee names in the order of their eligibility for promotion
Ramu
Bimu
Somu
Gomu
Vimu
The current size of the collection is 8
The size after removing the extra space is 5
```

- c. Hint – List multiples the memory when we add elements, ensure you use only the size that is equal to the number of elements that are present.
- 4) Use the application created for question 1 and in the same class do the following
 - a. The need for the list is over as all the employees are promoted. Not print all the employee names in ascending order.
 - b. Example Input:

```
Please enter the employee names in the order of their eligibility for promotion
```

Ramu
Bimu
Somu
Gomu
Vimu
Promoted employee list:
Bimu
Gomu
Ramu
Somu
Vimu

Medium

- 1) Create an application that will take employee details (Use the employee class) and store it in a collection
 - a. The collection should be able to give back the employee object if the employee id is provided.
 - i. Hint – Use a collection that will store key-value pair.
 - b. The ID of employee can never be null or have duplicate values.
- 2) Use the application created for question 1. Store all the elements in the collection in a list.
 - a. Sort the employees based on their salary.
 - i. Hint – Implement the IComparable interface in the Employee class.
 - b. Given an employee id find the employee and print the details.
 - i. Hint – Use a LINQ with a where clause.`
- 3) Use the application created for question 2. Find all the employees with the given name (Name to be taken from user)
- 4) Use the application created for question 3. Find all the employees who are elder than a given employee (Employee given by user)

Hard

- 1) Use the application created in Question 1 of medium.
 - a. Display a menu to user which will enable to print all the employee details, add an employee, modify the details of an employee (all except id), print an employee details given his id and delete an employee from the collection
 - b. Ensure the application does not break at any point. Handles all the cases with proper response
 - i. Example – If user enters an employee id that does not exists the response should inform the user the same.