Day 7 Morning Assignment

By Manoj Karnatapu - NBHealthCareTechnologies

Assignment 1

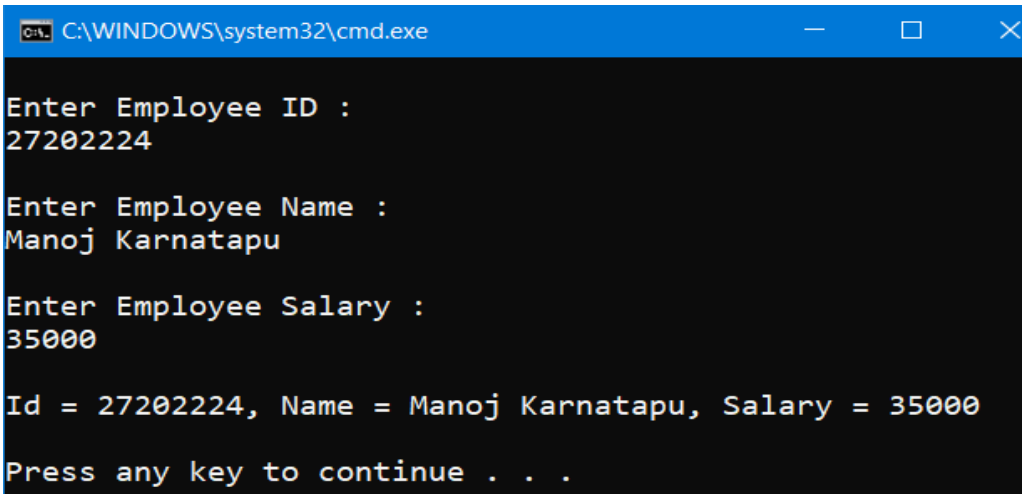
Write a C# Code, Create Employee class with three variables and two methods. Create an object and Call Methods.

Code

```
using System;

// Author : Manoj.Karnatapu
// Purpose : Creating an Emplyee Class with 2 Methods(ReadEmployee & PrintEmployee) &
             Calling the methods.
namespace Day7Project1
{
    class Employee
    {
        public int id;
        public string name;
        public int salary;

        // ReadEmployee Method to read the data from the user
        public void ReadEmployee()
        {
            Console.WriteLine("\nEnter Employee ID : ");
            id = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("\nEnter Employee Name : ");
            name = Console.ReadLine();

            Console.WriteLine("\nEnter Employee Salary : ");
            salary = Convert.ToInt32(Console.ReadLine());
        }
        // PrintEmployee Method will print the data of the Employee
        public void PrintEmployee()
        {
            Console.WriteLine($"\nId = {id}, Name = {name}, Salary = {salary}");

        }
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            Employee emp1 = new Employee();

            emp1.ReadEmployee();
            emp1.PrintEmployee();

            Console.ReadLine();
        }
```

```
    }
}
```

Output



Assignment 2

What are the 3 Definitions of class & 4 points about objects.

Answer

A: Class:

- A Class is a Collection of Variables and Methods
- A Class is like a Design/Blueprint to create objects.
- A Class consists of State and Behaviour. In which State is Nothing but Variables & Behaviour deals with Methods inside the class.
  i.e., state changes according to the behaviour (methods can modify the values of the variables.)

Objects:

- An Object is an instance of a class.
- We can Create any Number of objects.
- Objects occupy memory when we create.
- Objects are basically of reference types, because they store the address of the data present inside an object in Stack Memory & the actual data is stored in the Heap Memory

Assignment 3

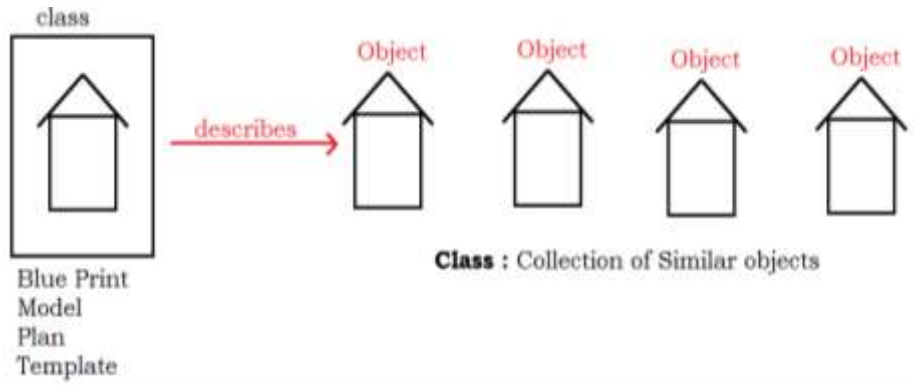
Pictorial representation of Class and Multiple Objects.

Answer

A:

Class:

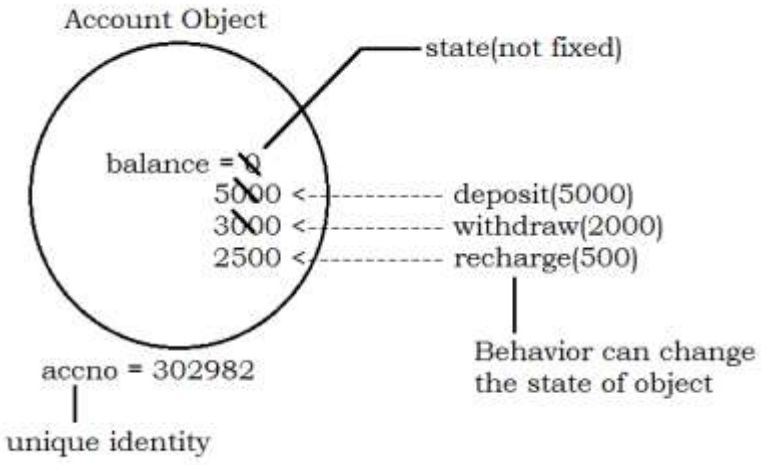
A Class is the complete representation of Object. It's Like a Blue Print to the Objects. In the Below picture, each object is considered as Account Object created using Class (Model/Blue Print). In which each object consists of different variables and methods.



Objects:

In a Banking Application, we have multiple Objects inside a class. Where as if you consider the Account Object, it consists of variables like balance & methods like Deposit(), WithDraw(), Recharge().

State is a variable; it changes according to the usage of methods. Behaviour (Methods) can change the state of an object. Whereas identity, it is unique and it is used to access the object.



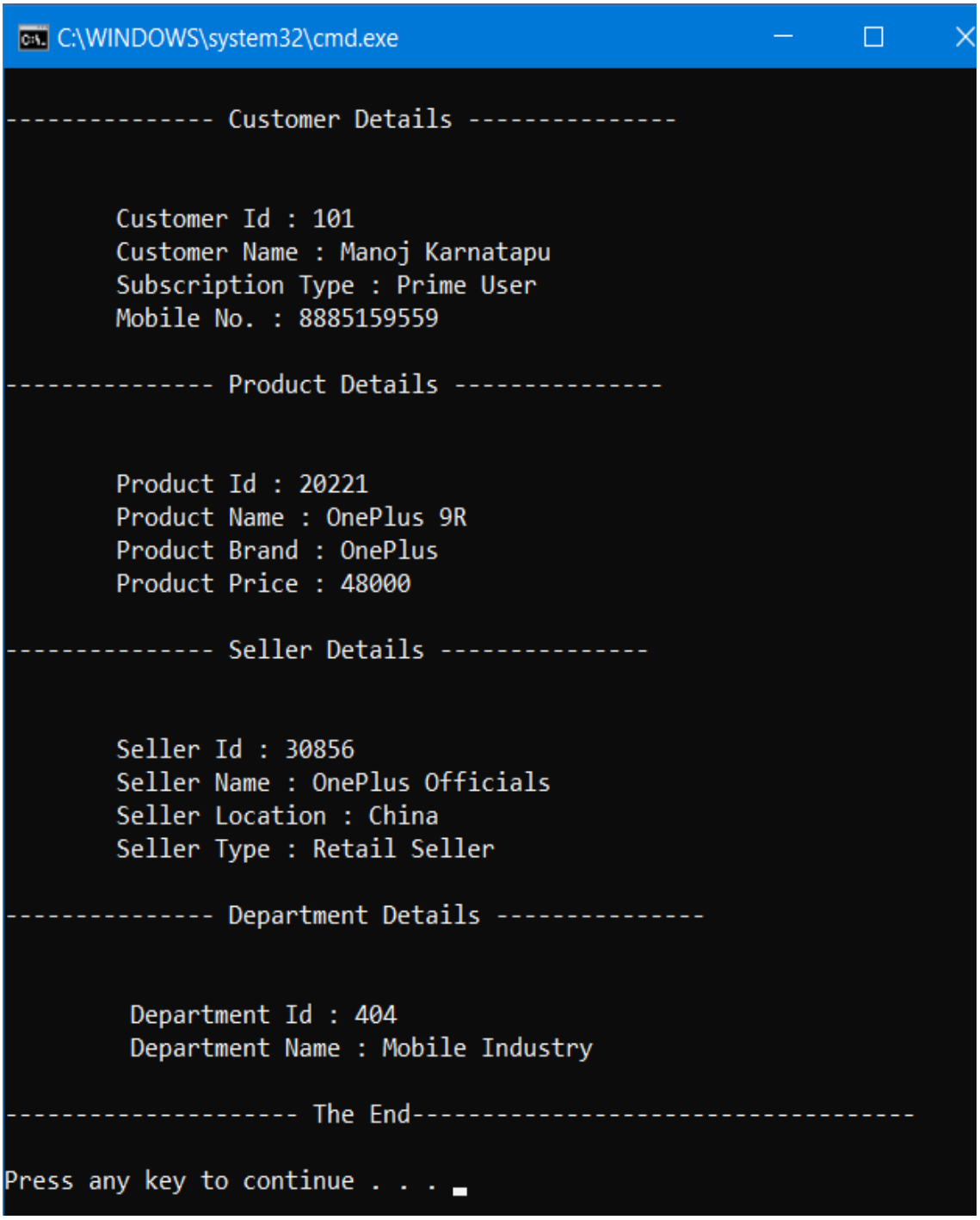
Assignment 4

Create a C# Code with 4 Classes (Customer, Product, Seller & Department) & Read and Print the Data by Calling Methods.

Code

```
using System;
using System.Collections.Generic;

// Author : Manoj.Karnatapu
// Purpose : Creating an 4 different Class with Required 2 Methods(ReadingData & PrintData)
& Calling the methods.

// Reference : To Check the Reference of the code. Go through the Day7Project2 folder for
complete Code.
namespace Day7Project2
{
    internal class Program
    {
        // Customer Class
        // Products Class
        // Seller Class
        // Department Class
        static void Main(string[] args)
        {
            // Customer Instance
            Customer customer1 = new Customer();
            Console.WriteLine("Enter Customer Details : ");
            customer1.CreateCustomerData();

            // Products Instance
            Products product1 = new Products();
            Console.WriteLine("Enter Product Details : ");
            product1.CreateProductData();

            // Seller instance
            Seller sell1 = new Seller();
            Console.WriteLine("Enter Seller Details : ");
            sell1.CreateSellerData();

            // Department instance
            Department dep1 = new Department();
            Console.WriteLine("Enter Department Details : ");
            dep1.ReadDepartment();

            customer1.DisplayCustomerProfile();

            product1.DisplayProducts();

            sell1.DisplaySellerData();

            dep1.DisplayDepartment();
            Console.WriteLine("\n---------------------- The End-----------------------");

            Console.ReadLine();
        }
    }
}
```

Output


C:\WINDOWS\system32\cmd.exe
---------- Customer Details ----------
Customer Id : 101
Customer Name : Manoj Karnatapu
Subscription Type : Prime User
Mobile No. : 8885159559
---------- Product Details ----------
Product Id : 20221
Product Name : OnePlus 9R
Product Brand : OnePlus
Product Price : 48000
---------- Seller Details ----------
Seller Id : 30856
Seller Name : OnePlus Officials
Seller Location : China
Seller Type : Retail Seller
---------- Department Details ----------
Department Id : 404
Department Name : Mobile Industry
---------- The End----------
Press any key to continue . . .

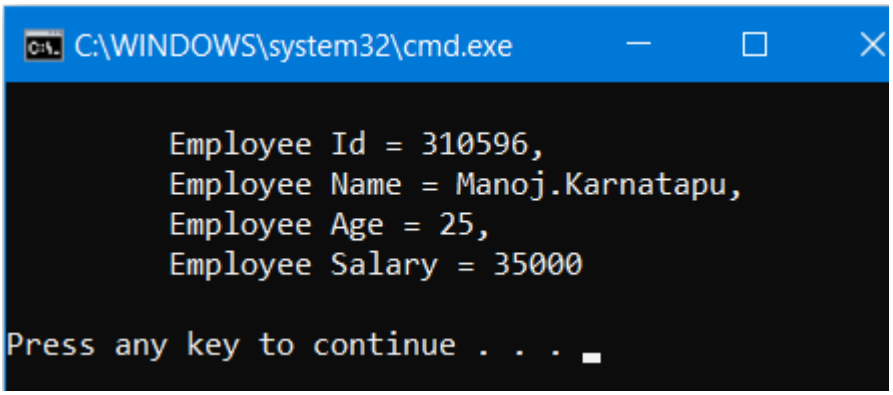
Assignment 5

Create Employee Class with 3 Public variables. Crete object and initialize with values while creating object and print the values.

Code

```
using System;

// Author : Manoj.Karnatapu
// Purpose : Create Employee Class with 3 Public variables. Crete object and initialize
with values while creating object and print the values.

namespace Day7Project3
{
    class Employee
    {
        public int employeeId;
        public string employeeName;
        public int employeeAge;
        public int employeeSalary;
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Employee emp = new Employee() {employeeId = 310596, employeeName =
"Manoj.Karnatapu", employeeAge = 25,
             employeeSalary = 35000};

            Console.WriteLine($"\n\t Employee Id = {emp.employeeId},\n\t Employee Name =
{emp.employeeName}," +
                $"\n\t Employee Age = {emp.employeeAge},\n\t Employee Salary =
{emp.employeeSalary}");

            Console.ReadLine();
        }
    }
}
```

Output



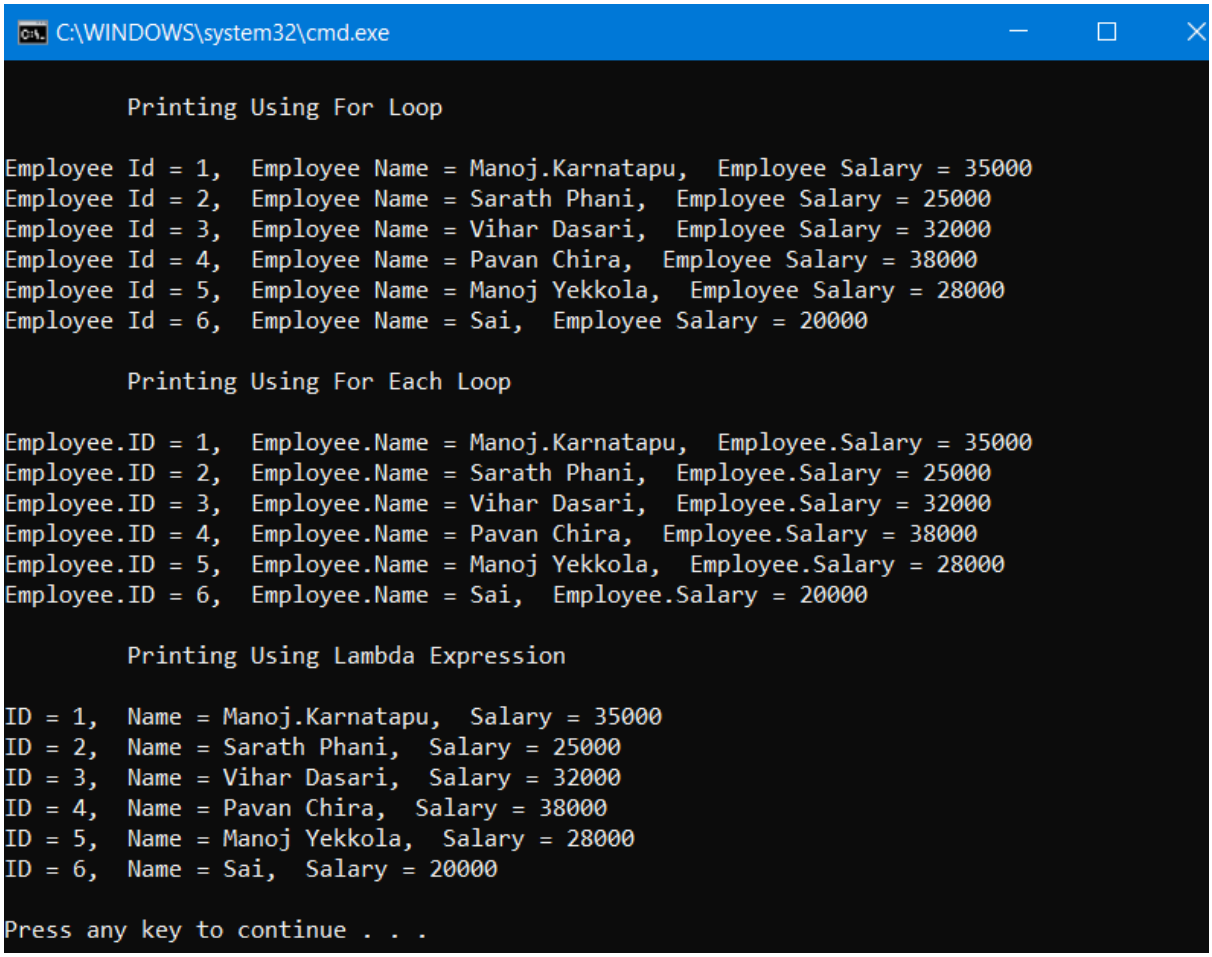
Assignment 6

Create Employee class, with Employee's array object and initialize with 5 employees. Print Output using for Loop, foreach loop & Lambda expression.

Code

```
using System;
using System.Linq;

// Author : Manoj.Karnatapu
// Purpose : Create Employee class, with Employee's array object and initialize with 5
employees. Print Output using for Loop, foreach loop & Lambda expression.

namespace Day7Project4
{
    internal class Program
    {
        class Employee
        {
            public int empId;
            public string empName;
            public int empSalary;
        }
        static void Main(string[] args)
        {
            Employee[] employees = new Employee[]
            {
                new Employee(){empId = 1, empName = "Manoj.Karnatapu", empSalary = 35000},
                new Employee(){empId = 2, empName = "Sarath Phani", empSalary = 25000},
                new Employee(){empId = 3, empName = "Vihar Dasari", empSalary =32000},
                new Employee(){empId = 4, empName = "Pavan Chira", empSalary = 38000},
                new Employee(){empId = 5, empName = "Manoj Yekkola", empSalary = 28000},
                new Employee(){empId = 6, empName = "Sai", empSalary = 20000}
            };

            // Using For Loop
            Console.WriteLine("\n\t Printing Using For Loop\n");
            for (int i = 0; i < employees.Length; i++)
            {
                Console.WriteLine($"Employee Id = {employees[i].empId},  Employee Name =
{employees[i].empName},  Employee Salary = {employees[i].empSalary}");
            }

            // Using For Each Loop
            Console.WriteLine("\n\t Printing Using For Each Loop\n");
            foreach(var e in employees)
            {
                Console.WriteLine($"Employee.ID = {e.empId},  Employee.Name = {e.empName},
Employee.Salary = {e.empSalary}");
            }
            // Using Lambda Expression
            Console.WriteLine("\n\t Printing Using Lambda Expression\n");

            employees.ToList().ForEach(d => Console.WriteLine($"ID = {d.empId},  Name =
{d.empName},  Salary = {d.empSalary}"));

            Console.ReadLine();
        }
    }
}
```

Output:



Assignment 7

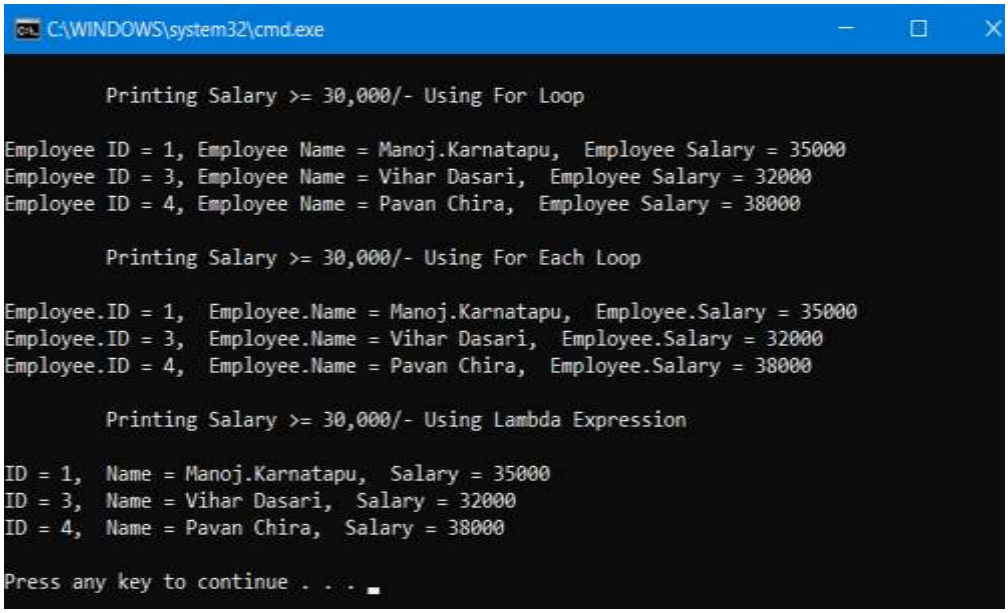
With Reference to Assignment 6, write C# code to print employees who is getting salary >= 5000 using for loop, foreach loop & Lambda expression.

Code

```
using System;
using System.Linq;

// Author : Manoj.Karnatapu
// Purpose : Create Employee class, with Employee's array object and initialize with 5 employees & write C# code to print employees who is getting salary >= 30,000 using for loop, foreach loop & Lambda expression.

namespace Day7Project5
{
    internal class Program
    {
        class Employee
        {
            public int empId;
            public string empName;
            public int empSalary;
        }
        static void Main(string[] args)
```

```
        {
            Employee[] employees = new Employee[]
            {
                new Employee(){empId = 1, empName = "Manoj.Karnatapu", empSalary = 35000},
                new Employee(){empId = 2, empName = "Sarath Phani", empSalary = 25000},
                new Employee(){empId = 3, empName = "Vihar Dasari", empSalary =32000},
                new Employee(){empId = 4, empName = "Pavan Chira", empSalary = 38000},
                new Employee(){empId = 5, empName = "Manoj Yekkola", empSalary = 28000},
                new Employee(){empId = 6, empName = "Sai", empSalary = 20000}
            };

            // Using For Loop
            Console.WriteLine("\n\t Printing Salary >= 30,000/- Using For Loop\n");
            for (int i = 0; i < employees.Length; i++)
            {
                if(employees[i].empSalary >= 30000)
                    Console.WriteLine($"Employee ID = {employees[i].empId}, Employee Name =
{employees[i].empName},  Employee Salary = {employees[i].empSalary}");
            }

            // Using For Each
            Console.WriteLine("\n\t Printing Salary >= 30,000/- Using For Each Loop\n");
            foreach (var e in employees)
            {
                if(e.empSalary >= 30000)
                    Console.WriteLine($"Employee.ID = {e.empId},  Employee.Name =
{e.empName},  Employee.Salary = {e.empSalary}");
            }

            // Using Lambda Expression
            Console.WriteLine("\n\t Printing Salary >= 30,000/- Using Lambda
Expression\n");
            employees.ToList().Where(e => e.empSalary >= 30000).ToList().ForEach(e =>
Console.WriteLine($"ID = {e.empId},  Name = {e.empName},  Salary = {e.empSalary}"));

            Console.ReadLine();
        }
    }
}
```

Output



Assignment 8

Similar to 6 and 7, Create a list of Customer and Product Arrays, using for loop, foreach & Lambda

Code

```
using System;
using System.Linq;

// Author : Manoj.Karnatapu
// Purpose : Create Customer and Products class data in the form of Array while creating Object & print using For loop, For Each loop and Lambda Expressions.

namespace Day7Project6
{
    class Customer
    {
        public int customerId;
        public string customerName;
        public string customerSubscription;
    }

    class Products
    {
        public int productId;
        public string productName;
        public int productPrice;
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Customer[] customers = new Customer[]
            {
                new Customer(){ customerId = 1, customerName = "Manoj.Karnatapu", customerSubscription = "Prime User"},
                new Customer(){ customerId = 2, customerName = "Pavan Kumar", customerSubscription = "General"},
                new Customer(){ customerId = 3, customerName = "Vihar Dasari", customerSubscription = "Prime User"}
            };
            // Using For Loop
            Console.WriteLine("\n\t Printing Customer Data Using For Loop\n");
            for (int i = 0; i < customers.Length; i++)
            {
                Console.WriteLine($"Customer Id = {customers[i].customerId},  Customer Name = {customers[i].customerName},  Customer Subscription = {customers[i].customerSubscription}");
            }

            // Using For Each Loop
            Console.WriteLine("\n\t Printing Customer Data Using For Each Loop\n");
            foreach (var e in customers)
            {
                Console.WriteLine($"Customer.ID = {e.customerId},  Customer.Name= {e.customerName},  Customer.Subscription = {e.customerSubscription}");
            }
            // Using Lambda Expression
            Console.WriteLine("\n\t Printing Customer Data Using Lambda Expression\n");

            customers.ToList().ForEach(d => Console.WriteLine($"ID = {d.customerId},  Name = {d.customerName},  Subscription = {d.customerSubscription}"));


Console.WriteLine("\n=====================================================================\n");

            // For Products Class
```

```
            Products[] products = new Products[]
            {
                new Products(){ productId = 1, productName = "Nokia 5500", productPrice = 10000},
                new Products(){ productId = 2, productName = "RedMi Note 9 pro", productPrice =
13000},
                new Products(){ productId = 3, productName = "OnePlus 9R", productPrice = 48000}
            };
            // Using For Loop
            Console.WriteLine("\n\t Printing Products Data Using For Loop\n");
            for (int i = 0; i < products.Length; i++)
            {
                Console.WriteLine($"Product Id = {products[i].productId},  Product Name =
{products[i].productName},  Product price = {products[i].productPrice}");
            }
            // Using For Each Loop
            Console.WriteLine("\n\t Printing Products Data Using For Each Loop\n");
            foreach (var p in products)
            {
                Console.WriteLine($"Product.ID = {p.productId},  Product.Name= {p.productName},
Product.Price = {p.productPrice}");
            }
            // Using Lambda Expression
            Console.WriteLine("\n\t Printing Products Data Using Lambda Expression\n");

            products.ToList().ForEach(d => Console.WriteLine($"ID = {d.productId},  Name =
{d.productName},  Price = {d.productPrice}"));

            Console.ReadLine();

        }
    }
}
```

Output