

Day7 Morning Assignment

By

Narala Praveen

01-Feb-2022

Question 1:

Create Employee Class with three variables and two methods.

Read Employee and Print Employee and create an object and call methods?

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _7dayMorningproject1
{
    //*****\\
    //Author:Narala Praveen
    //Purpose:To Create Employee class with three variables and two methods
    //Read Employee and PrintEmployee And Create an object and call methods
    //*****\\
    class Employee
    {
        private int Id;
        private string Name;
        private int salary;

        public void ReadEmployee()
        {
            Console.WriteLine("Enter Id:");
            Id = Convert.ToInt32(Console.ReadLine());

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

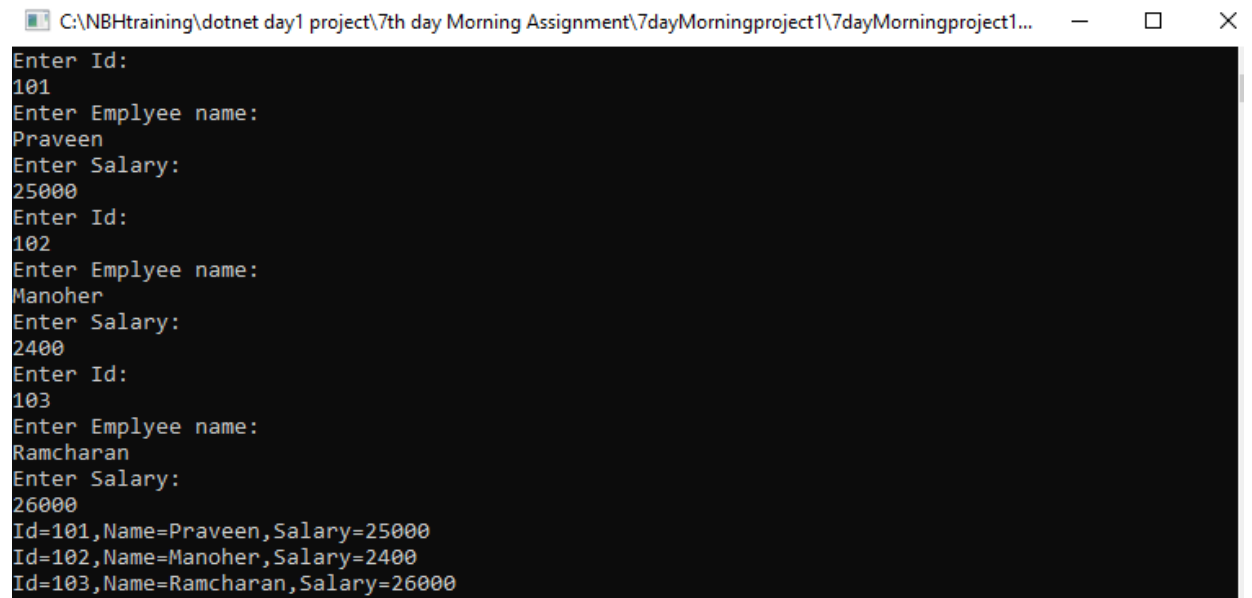
            Console.WriteLine("Enter Salary:");
            salary = Convert.ToInt32(Console.ReadLine());
        }
        public void PrintEmployee()
        {
            Console.WriteLine($"Id={Id},Name={Name},Salary={salary}");
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Employee emp1 = new Employee();
            Employee emp2 = new Employee();
            Employee emp3 = new Employee();

            emp1.ReadEmployee();
            emp2.ReadEmployee();
            emp3.ReadEmployee();

            emp1.PrintEmployee();
            emp2.PrintEmployee();
            emp3.PrintEmployee();

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

Output:



```
C:\NBHtraining\dotnet day1 project\7th day Morning Assignment\7dayMorningproject1\7dayMorningproject1...
Enter Id:
101
Enter Employee name:
Praveen
Enter Salary:
25000
Enter Id:
102
Enter Employee name:
Manoher
Enter Salary:
2400
Enter Id:
103
Enter Employee name:
Ramcharan
Enter Salary:
26000
Id=101,Name=Praveen,Salary=25000
Id=102,Name=Manoher,Salary=2400
Id=103,Name=Ramcharan,Salary=26000
```

Question2:

Write the 3 definitions of class and 4 points about object discussed in the class

Class definitions:

Definition1: “A group of Variables and Methods”.

Definition2:” Class is like design/blueprint to create an object”.

Definition3:” A class consists of State(Variables) and Behaviour(Methods)”.

Objects:

Point1: “An object is an instance of a class “.

Point2: “We can create any number of objects”.

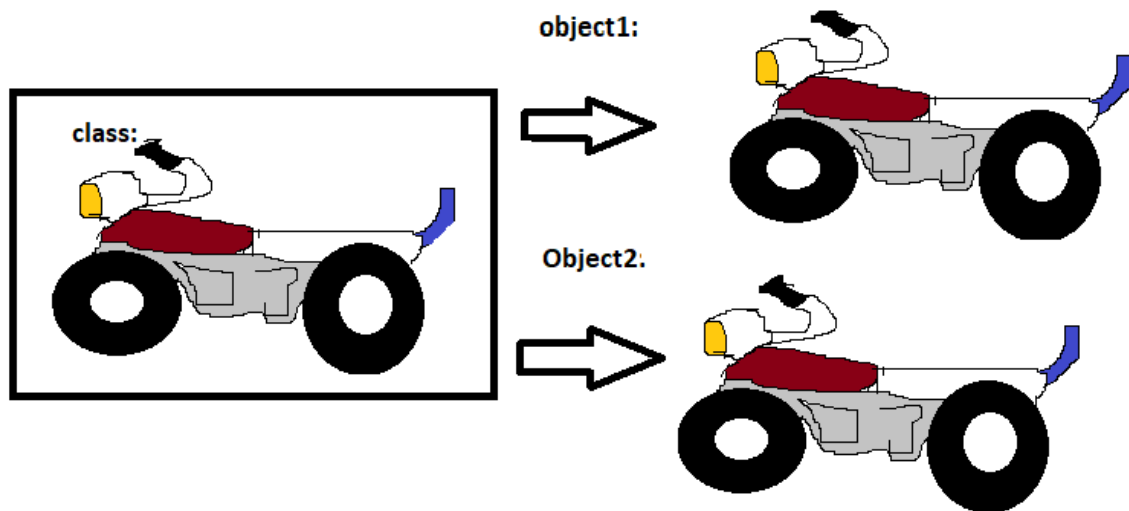
Point4:” object occupy memory”.

Point5:”Objects are reference type”.

Questions3 :

Pictorially represent class and Multiple objects?

Picture:



Question 4:

Create below classes:

1.Customer

2.Product

3.Seller

4. Department.

```
Code for product class: using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day7project2
{
    //*****\\
    //Author :Narala Praveen
    //Purpose:Creating Product class
    //*****\\
    class Product
    {
        private string Name;
        private int price;
        private string colour;

        public void ReadProduct()
        {
            Console.WriteLine("Enter Product name:");
            Name= Console.ReadLine();

            Console.WriteLine("Enter Product Price");
            price= Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("Enter Product colour");
            colour= Console.ReadLine();

        }
        public void PrintProduct()
        {
            Console.WriteLine($"Name={Name},Price={price},Colour={colour}");
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Product p1= new Product();
            Product p2= new Product();
            Product p3= new Product();

            p1.ReadProduct();
            p2.ReadProduct();
            p3.ReadProduct();

            p1.PrintProduct();
            p2.PrintProduct();
            p3.PrintProduct();

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

```
}  
    }  
}
```

C:\NBHtraining\dotnet day1 project... — □ ×

```
Enter Product name:  
sanitizer  
Enter Product Price  
300  
Enter Product colour  
blue  
Enter Product name:  
mask  
Enter Product Price  
40  
Enter Product colour  
black  
Enter Product name:  
handgloves  
Enter Product Price  
30  
Enter Product colour  
white  
Name=sanitizer,Price=300,Colour=blue  
Name=mask,Price=40,Colour=black  
Name=handgloves,Price=30,Colour=white
```

Output:

Code for Customer Class:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace day7project2a
{
    //*****\\
    //Author:Narala Praveen
    //Purpose:create class for customer
    class Customer
    {
        private string Name;
        private int ID;
        private int age;

        public void ReadCustomer()
        {
            Console.WriteLine("Enter Name:");
            Name= Console.ReadLine();

            Console.WriteLine("Enter ID:");
            ID=Convert.ToInt32(Console.ReadLine());

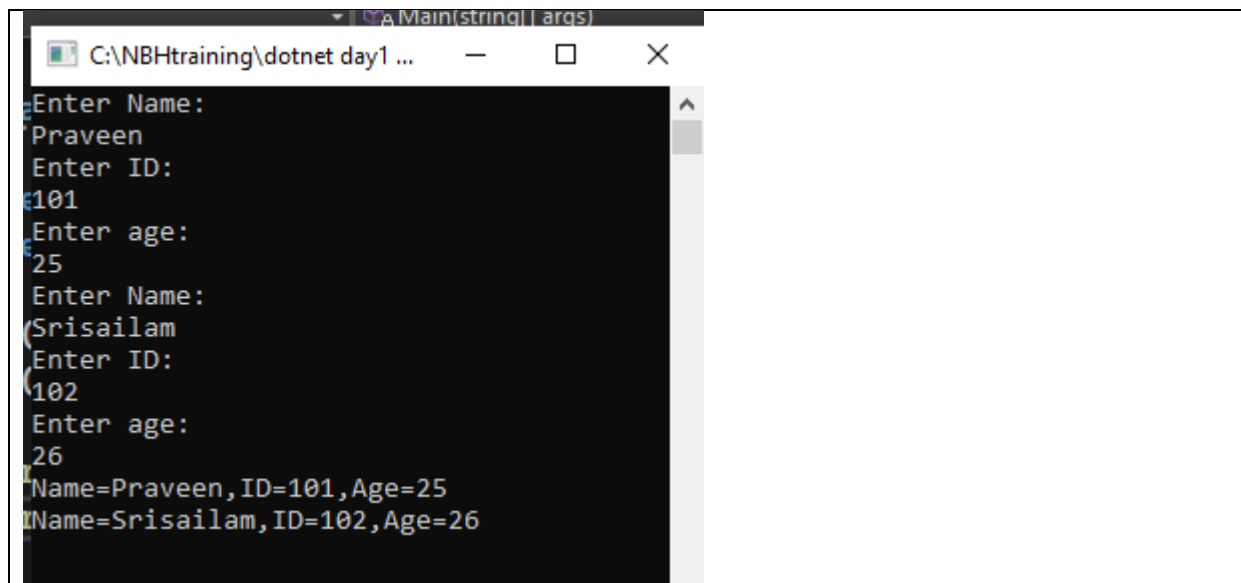
            Console.WriteLine("Enter age:");
            age= Convert.ToInt32(Console.ReadLine());
        }
        public void PrintCustomer()
        {
            Console.WriteLine($"Name={Name},ID={ID},Age={age}");
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Customer customer1 = new Customer();
            Customer customer2 = new Customer();

            customer1.ReadCustomer();
            customer2.ReadCustomer();

            customer1.PrintCustomer();
            customer2.PrintCustomer();

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

Output:



The image shows a screenshot of a Windows console application window. The title bar at the top reads "C:\NBHtraining\dotnet day1 ..." followed by standard window controls. The console output is as follows:

```
Enter Name:
Praveen
Enter ID:
101
Enter age:
25
Enter Name:
Srisailam
Enter ID:
102
Enter age:
26
Name=Praveen,ID=101,Age=25
Name=Srisailam,ID=102,Age=26
```

Class Seller:

```
Code: using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace day7project2seller
{
    //*****\\
    //Author:Narala Praveen
    //Purpose: To create Seller class
    //*****\\
    class Seller
    {
        private int ID;
        private string Name;
        private string Brand;

        public void ReadSeller()
        {
            Console.WriteLine("Enter ID:");
            ID = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("enter Name:");
            Name= Console.ReadLine();

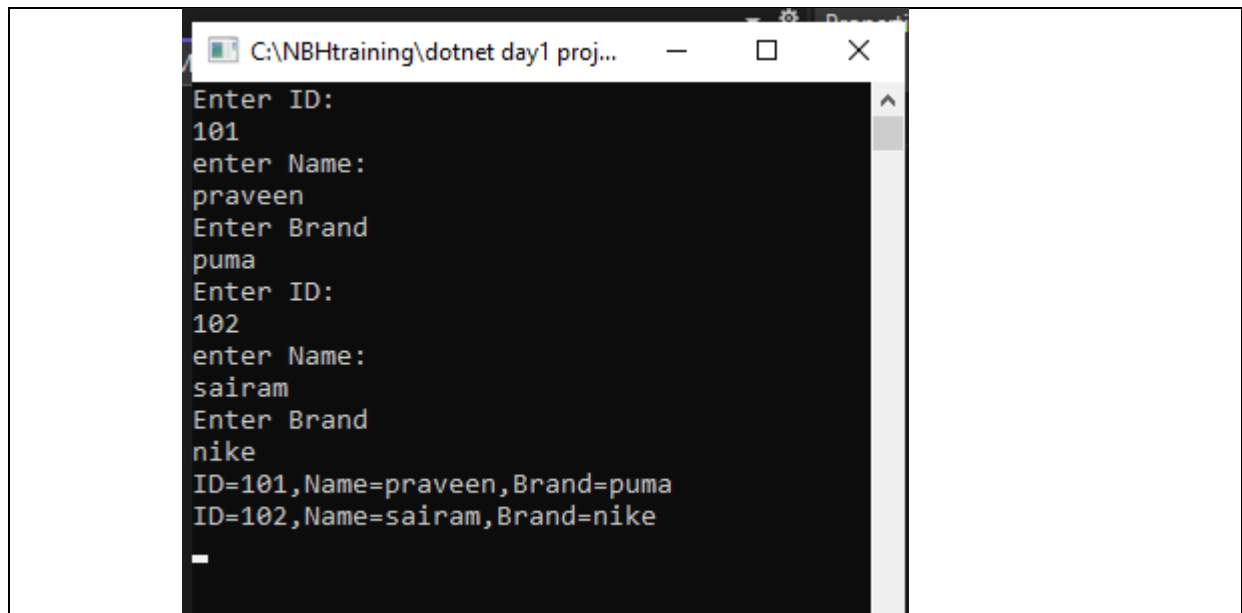
            Console.WriteLine("Enter Brand");
            Brand= Console.ReadLine();

        }
        public void PrintSeller()
        {
            Console.WriteLine($"ID={ID},Name={Name},Brand={Brand}");
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Seller seller1 = new Seller();
            Seller seller2 = new Seller();
            seller1.ReadSeller();
            seller2.ReadSeller();

            seller1.PrintSeller();
            seller2.PrintSeller();

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

Output:



```
C:\NBHtraining\dotnet day1 proj...
Enter ID:
101
enter Name:
praveen
Enter Brand
puma
Enter ID:
102
enter Name:
sairam
Enter Brand
nike
ID=101,Name=praveen,Brand=puma
ID=102,Name=sairam,Brand=nike
_
```

Create Department Class:

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day7project2Department
{
    //*****\\
    //Author :Narala Praveen//
    //Purpose:To create Department Class//
    //*****\\
    class Department
    {
        private string Name;
        private int Number;
        private string Description;

        public void ReadDepartment()
        {
            Console.WriteLine("Enter Name:");
            Name = Console.ReadLine();

            Console.WriteLine("Enter Department number:");
            Number = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("Enter Description:");
            Description = Console.ReadLine();
        }

        public void PrintDepartment()
        {
            Console.WriteLine($"Name={Name}, Number={Number}, Description={Description}");
        }
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            Department department1 = new Department();
            Department department2 = new Department();

            department1.ReadDepartment();
            department2.ReadDepartment();

            department1.PrintDepartment();
            department2.PrintDepartment();

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

Output:

Lifecycle Events Thread: Stack Frame:

C:\NBHtraining\dotnet day1 project\7th day Morning Assignment\Day7project2Department

Enter Name:

police

Enter Department number:

101

Enter Description:

subinspector

Enter Name:

ram

Enter Department number:

102

Enter Description:

constable

Name=police,Number=101,Description=subinspector

Name=ram,Number=102,Description=constable

Question 5:

Create Employee class with 3 public variables create employee object and initialize with values creating object and print with values?

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day7Project5
{
    //*****\\
    //Author:Narala Praveen
    //Purpose:To Create Employee class with Public Vaariables
    //*****\\
    class Employee
    {
        public string Name;
        public int ID;
        public int Salary;

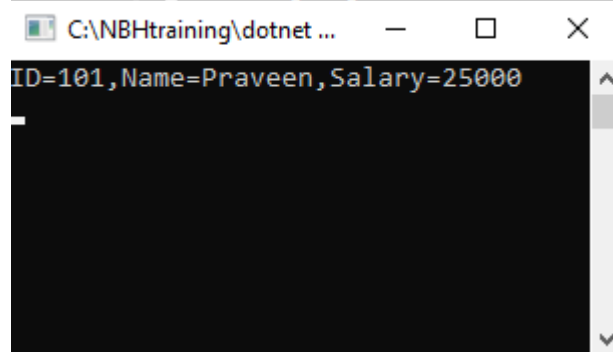
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            Employee emp1 = new Employee() { ID = 101, Name = "Praveen", Salary =
25000 };

            Console.WriteLine($"ID={emp1.ID},Name={emp1.Name},Salary={emp1.Salary}");
            Console.ReadLine();

        }
    }
}
```

Output:



Question6:

Create Employee class as shown below:

Class Employee

```
{  
Public int id ;  
Public string name;  
Public int salary;  
}
```

Now Create employees array object and initialize with 5 employees

Write code using

A) For loop

B) For each loop

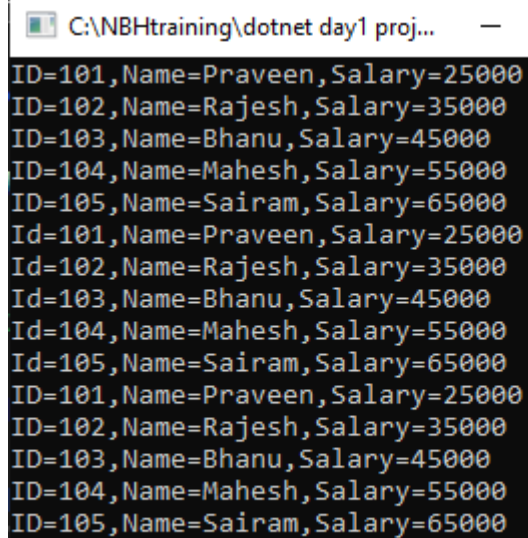
C) Lambda expression.

Code:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace Day7Project6  
{  
    //*****\\  
    //Author:Narala Praveen  
    //Purpose:Create employee array object and initialize with 5  
employees  
    //*****\\  
    class Employee  
    {  
        public int id;  
        public string name;  
        public int salary;  
    }  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
            Employee[] employees = new Employee[]  
            {  
                new Employee() { id = 101, name = "Praveen", salary = 25000 },  
                new Employee() { id = 102, name = "Rajesh", salary = 35000 },  
                new Employee() { id = 103, name = "Bhanu", salary = 45000 },  
                new Employee() { id = 104, name = "Mahesh", salary = 55000 },  
                new Employee() { id = 105, name = "Sairam", salary = 65000 },  
            };  
  
            //For loop  
            for(int i = 0; i < employees.Length; i++)  
  
Console.WriteLine($"ID={employees[i].id},Name={employees[i].name},Salary={employee  
es[i].salary}");  
  
            //For each loop  
  
            foreach(var e in employees)  
            {  
                Console.WriteLine($"Id={e.id},Name={e.name},Salary={e.salary}");  
            }  
        }  
    }  
}
```

```
    }  
  
    //Lambda Expression  
  
    employees.ToList().ForEach(e =>  
Console.WriteLine($"ID={e.id},Name={e.name},Salary={e.salary}"));  
  
    Console.ReadLine();  
}  
  
}  
  
}
```

Output:



```
C:\NBHtraining\dotnet day1 proj...  
ID=101,Name=Praveen,Salary=25000  
ID=102,Name=Rajesh,Salary=35000  
ID=103,Name=Bhanu,Salary=45000  
ID=104,Name=Mahesh,Salary=55000  
ID=105,Name=Sairam,Salary=65000  
Id=101,Name=Praveen,Salary=25000  
Id=102,Name=Rajesh,Salary=35000  
Id=103,Name=Bhanu,Salary=45000  
Id=104,Name=Mahesh,Salary=55000  
Id=105,Name=Sairam,Salary=65000  
ID=101,Name=Praveen,Salary=25000  
ID=102,Name=Rajesh,Salary=35000  
ID=103,Name=Bhanu,Salary=45000  
ID=104,Name=Mahesh,Salary=55000  
ID=105,Name=Sairam,Salary=65000
```


Question7:

For the above project write code to print employees who is getting salary ≥ 5000 using

a)for loop

b)for each loop

c)Lambda Expression

```
Code: using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day7Project7
{
    //*****\\
    //Author:Narala Praveen
    //Purpose:To print employees whose salary equal and morethan 50000
    //*****\\
    class Employee
    {
        public int id;
        public string name;
        public int salary;
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Employee[] employees = new Employee[]
            {
                new Employee() { id = 101, name = "Praveen", salary = 25000 },
                new Employee() { id = 102, name = "Rajesh", salary = 35000 },
                new Employee() { id = 103, name = "Bhanu", salary = 45000 },
                new Employee() { id = 104, name = "Mahesh", salary = 55000 },
                new Employee() { id = 105, name = "Sairam", salary = 65000 },
            };

            //For loop
            for (int i = 0; i < employees.Length; i++)
            {
                if(employees[i].salary>=50000)

Console.WriteLine($"ID={employees[i].id},Name={employees[i].name},Salary={employees[i].salary}");
            }
            //For each loop

            foreach (var e in employees)
            {
                if(e.salary>=50000)
                Console.WriteLine($"Id={e.id},Name={e.name},Salary={e.salary}");
            }

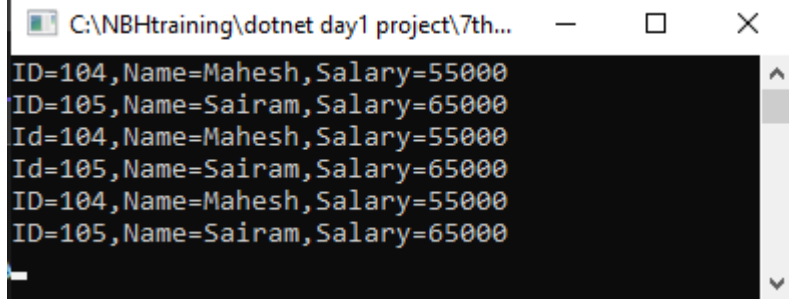
            //Lambda Expression

            employees.ToList().Where(e=>e.salary>=50000).ToList().ForEach(e =>
Console.WriteLine($"ID={e.id},Name={e.name},Salary={e.salary}"));

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

```
}  
    }  
}
```

Output:



```
C:\NBHtraining\dotnet day1 project\7th...  
ID=104,Name=Mahesh,Salary=55000  
ID=105,Name=Sairam,Salary=65000  
Id=104,Name=Mahesh,Salary=55000  
Id=105,Name=Sairam,Salary=65000  
ID=104,Name=Mahesh,Salary=55000  
ID=105,Name=Sairam,Salary=65000
```

Question 8:

Similar to 6 and 7 projects create list of Customer and Product Arrays and Practice

a) For Loop

b) For each loop

c) Lambda Expression

```
Customer code: using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day7Project8Customer
{
    //*****\\
    //Author:Narala Praveen
    //Purpose:To create customer class and declare array objects
    //*****\\
    class Customer
    {
        public string name;
        public int id;
        public int age;
    }
    internal class Program
    {
        static void Main(string[] args)
        {

            Customer[] customers = new Customer[]
            {
                new Customer() { id = 101, name = "Praveen", age = 25},
                new Customer() { id = 102, name = "Rajesh", age = 35},
                new Customer() { id = 103, name = "Bhanu", age = 45},
                new Customer() { id = 104, name = "Mahesh", age = 55},
                new Customer() { id = 105, name = "Sairam", age = 65},
            };

            //For loop
            for (int i = 0; i < customers.Length; i++)
            {

                Console.WriteLine($"ID={customers[i].id},Name={customers[i].name},Age={customers[i].age}");
            }
            //For each loop

            foreach (var c in customers )
            {

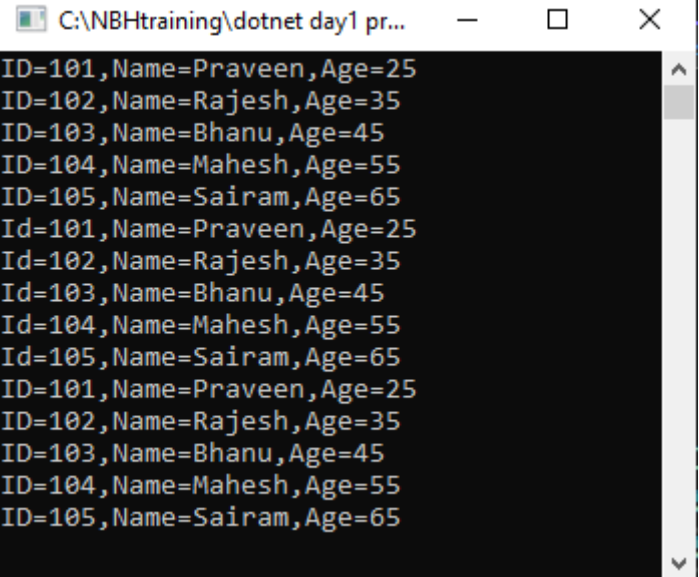
                Console.WriteLine($"Id={c.id},Name={c.name},Age={c.age}");
            }

            //Lambda Expression

            customers.ToList().ForEach(e =>
            Console.WriteLine($"ID={e.id},Name={e.name},Age={e.age}"));
        }
    }
}
```

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

Output:



The screenshot shows a Windows console window titled "C:\NBHtraining\dotnet day1 pr...". The output consists of 15 lines of text, each representing a record with an ID, Name, and Age. The first five lines are unique records, and the next ten lines are a repetition of these five records. The records are: ID=101, Name=Praveen, Age=25; ID=102, Name=Rajesh, Age=35; ID=103, Name=Bhanu, Age=45; ID=104, Name=Mahesh, Age=55; and ID=105, Name=Sairam, Age=65.

```
ID=101,Name=Praveen,Age=25  
ID=102,Name=Rajesh,Age=35  
ID=103,Name=Bhanu,Age=45  
ID=104,Name=Mahesh,Age=55  
ID=105,Name=Sairam,Age=65  
Id=101,Name=Praveen,Age=25  
Id=102,Name=Rajesh,Age=35  
Id=103,Name=Bhanu,Age=45  
Id=104,Name=Mahesh,Age=55  
Id=105,Name=Sairam,Age=65  
ID=101,Name=Praveen,Age=25  
ID=102,Name=Rajesh,Age=35  
ID=103,Name=Bhanu,Age=45  
ID=104,Name=Mahesh,Age=55  
ID=105,Name=Sairam,Age=65
```

Customer Age greater and equal to 50:

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day7Project8_CustomerAge
{
    //*****\\
    //Author:Narala Praveen
    //Purpose:To Print customer age greater than 50
    //*****\\
    class Customer
    {
        public string name;
        public int id;
        public int age;
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Customer[] customers = new Customer[]
            {
                new Customer() { id = 101, name = "Praveen", age = 25},
                new Customer() { id = 102, name = "Rajesh", age = 35},
                new Customer() { id = 103, name = "Bhanu", age = 45},
                new Customer() { id = 104, name = "Mahesh", age = 55},
                new Customer() { id = 105, name = "Sairam", age = 65},
            };

            //For loop
            for (int i = 0; i < customers.Length; i++)
            {
                if (customers[i].age > 50)

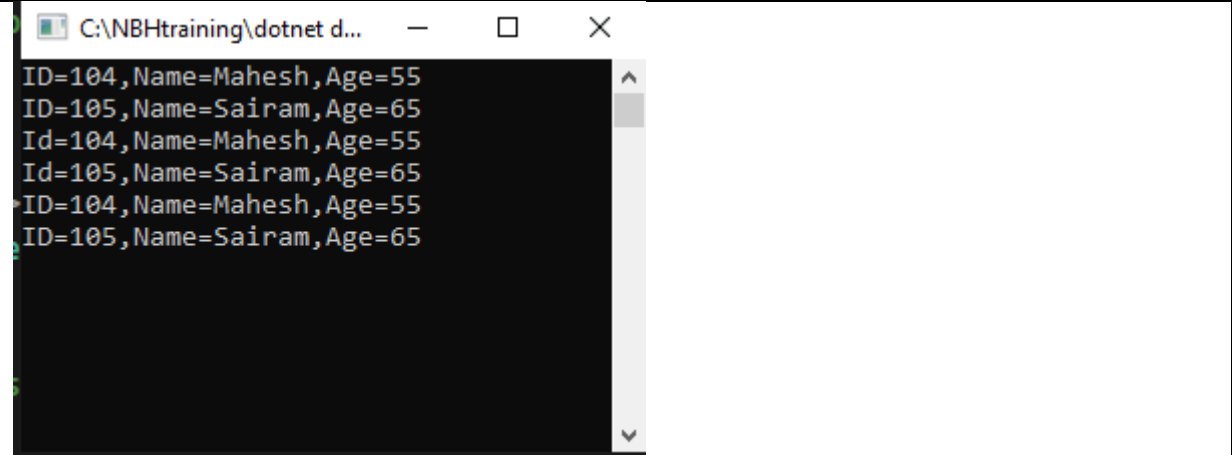
Console.WriteLine($"ID={customers[i].id},Name={customers[i].name},Age={customers[i].age}");
            }
            //For each loop
            foreach (var c in customers)
            {
                if (c.age > 50)
                    Console.WriteLine($"Id={c.id},Name={c.name},Age={c.age}");
            }

            //Lambda Expression

            customers.ToList().Where(c => c.age >= 50).ToList().ForEach(e =>
Console.WriteLine($"ID={e.id},Name={e.name},Age={e.age}"));

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

Output:



```
C:\NBHtraining\dotnet d...  
ID=104,Name=Mahesh,Age=55  
ID=105,Name=Sairam,Age=65  
Id=104,Name=Mahesh,Age=55  
Id=105,Name=Sairam,Age=65  
ID=104,Name=Mahesh,Age=55  
ID=105,Name=Sairam,Age=65
```

Class Product:

Code for product class and array objects declaration:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day7Project8_Products
{
    //*****\\
    //Author:Narala Praveen
    //Purpose:To create product class and declare array objects
    //*****\\
    class Product
    {
        public string name;
        public int price;
        public string colour;
    }
    internal class Program
    {
        static void Main(string[] args)
        {

            Product[] products = new Product[]
            {
                new Product() { colour = "black", name = "sanitizer", price = 25},
                new Product() { colour = "pink", name = "mask", price = 35},
                new Product() { colour = "red", name = "mouth fresher", price = 45},
                new Product() { colour = "white", name = "sheild", price = 55},
                new Product() { colour = "orange", name = "gloves", price = 65},
            };

            //For loop
            for (int i = 0; i < products.Length; i++)
            {

                Console.WriteLine($"Colour={products[i].colour},Name={products[i].name},price={products[i].price}");
            }
            //For each loop

            foreach(var p in products)
            {

                Console.WriteLine($"colour={p.colour},Name={p.name},price={p.price}");
            }

            //Lambda Expression

            products.ToList().ForEach(p =>
            Console.WriteLine($"colour={p.colour},Name={p.name},price={p.price}"));

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

Output:

Select C:\NBHtraining\dotnet day1 project\7th day Morning ,

```
Colour=black,Name=sanitizer,price=25
Colour=pink,Name=mask,price=35
Colour=red,Name=mouth fresher,price=45
Colour=white,Name=sheild,price=55
Colour=orange,Name=gloves,price=65
colour=black,Name=sanitizer,price=25
colour=pink,Name=mask,price=35
colour=red,Name=mouth fresher,price=45
colour=white,Name=sheild,price=55
colour=orange,Name=gloves,price=65
colour=black,Name=sanitizer,price=25
colour=pink,Name=mask,price=35
colour=red,Name=mouth fresher,price=45
colour=white,Name=sheild,price=55
colour=orange,Name=gloves,price=65
```


Class Product Price greater than 50

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day7Project8products_price
{
    //*****\\
    //Author:Narala Praveen
    //Purpose:to print the products whose price is greater and equal to 50
    class Product
    {
        public string name;
        public int price;
        public string colour;
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Product[] products = new Product[]
            {
                new Product() { colour = "black", name = "sanitizer", price = 25},
                new Product() { colour = "pink", name = "mask", price = 35},
                new Product() { colour = "red", name = "mouth fresher", price = 45},
                new Product() { colour = "white", name = "sheild", price = 55},
                new Product() { colour = "orange", name = "gloves", price = 65},
            };

            //For loop
            for (int i = 0; i < products.Length; i++)
            {
                if(products[i].price>=50)

Console.WriteLine($"Colour={products[i].colour},Name={products[i].name},price={pr
oducts[i].price}");
            }
            //For each loop

            foreach (var p in products)
            {
                if(p.price>=50)

Console.WriteLine($"colour={p.colour},Name={p.name},price={p.price}");
            }

            //Lambda Expression

            products.ToList().Where(p=>p.price>=50).ToList().ForEach(p =>
Console.WriteLine($"colour={p.colour},Name={p.name},price={p.price}"));

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

Output:

C:\NBHtraining\dotnet day1 project\7th day Morni...

```
Colour=white,Name=sheild,price=55  
Colour=orange,Name=gloves,price=65  
colour=white,Name=sheild,price=55  
colour=orange,Name=gloves,price=65  
colour=white,Name=sheild,price=55  
colour=orange,Name=gloves,price=65  
if(n.price>=50)
```