Day 8 Assignment
By
Triveni Anumolu
02-02-2022

1. Declare and initialize a list with 8 values. write for loop, foreach loop, lambda , linq query to print even numbers.

Code:

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

namespace Day8Project1
{
   internal class Program
   {
      //Author: Triveni Anumolu
      //Purpose: Declare & initialize a List With 8 values
      static void Main(string[] args)
      {
         List<int> data = new List<int>() { 74, 87, 99, 96, 44, 85, 34 };

         //Even Numbers Using For Loop
         for (int i = 0; i < data.Count; i++)
         {
            if (data[i] % 2 == 0)
               Console.WriteLine(data[i]);
         }

         //Even Numbers Using foreach loop
         foreach (var d in data)
         {
            if (d % 2 == 0)
               Console.WriteLine(d);
         }
         //Even Numbers using Lambda Expression
         data.Where(x => x % 2 == 0).ToList().ForEach(x => Console.WriteLine(x));


         //Even numbers using LINQ Query
         var result = from v in data
                      where v % 2 == 0
                      select v;
         result.ToList().ForEach(x => Console.WriteLine(x));
```

```
            Console.ReadLine();
        }

    }
}
```

Result:

```
74
96
44
34
74
96
44
34
74
96
44
34
74
96
44
34
```

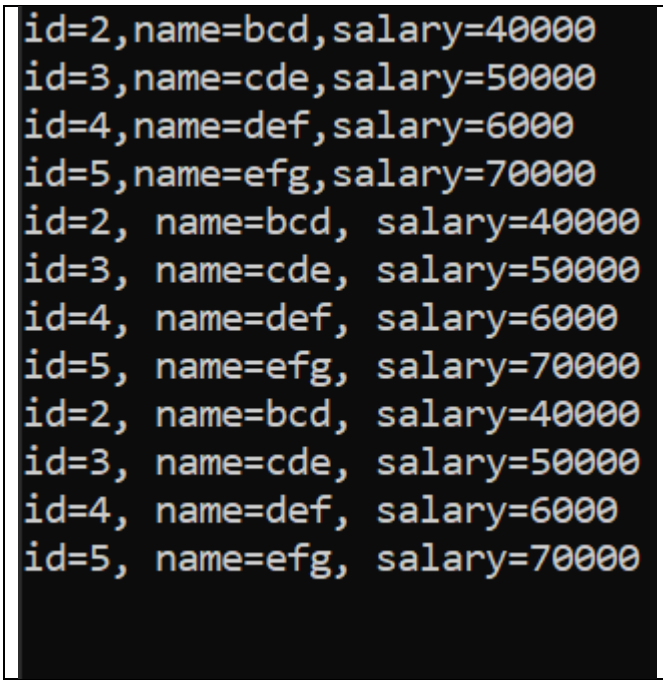
2. Create a class Employee with three variables as discussed in the class and create a list of Employees
public int id;
public string name;
public int salary;
write for loopforeach ,looplambda expression ,Linq query

Code:

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

namespace Day8Project2
    /*************************************************************************************
     Author: Triveni Anumolu
```

```
   Purpose:Creating employee array object and initializing using loops
   ***************************************************************************************/
   class Employee
   {
      public int id;
      public string name;
      public int salary;
   }
   class Program
   {
      static void Main(string[] args)
      {
         Employee[] emp = new Employee[]
         {
            new Employee(){id=1, name="abc",salary=2000 },
            new Employee(){id=2, name="bcd",salary=40000 },
            new Employee(){id=3, name="cde",salary=50000 },
            new Employee(){id=4, name="def",salary=6000 },
            new Employee(){id=5, name="efg",salary=70000 }
         };
         //for loop
         for(int i=0;i<emp.Length;i++)
         {
            if(emp[i].salary>5000)
               Console.WriteLine($"id={emp[i].id},name={emp[i].name},salary={emp[i].salary}");
         }
         //foreach loop
         foreach(var e in emp)
         {
            if(e.salary>5000)
            Console.WriteLine($"id={e.id}, name={e.name}, salary={e.salary}");
         }
         //lambda expression
         emp.ToList().Where(e=>e.salary>=5000).ToList().ForEach(e =>Console.WriteLine($"id={e.id},
name={e.name}, salary={e.salary}"));

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

Result:



3. Create a class Product and add variables id, name, price, brand print product (name and brand) whose price is more than 500using forforeach looplambdalinq query

Code:

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

namespace Day8Project3
{
   class Product
   //Author: Triveni anumolu
   //Purpose: Create a class Product and add variables id, name, price, brand. print product(name and brand) whose price is more than 500

   {
      public int Id;
      public string name;
      public int price;
      public string brand;
   }
   class Program
   {
      static void Main(string[] args)
```

```
    {
        List<Product> products = new List<Product>()
        {
            new Product() { Id = 101, name ="Air Conditioner", price = 63000, brand = "Daikin"},
            new Product() { Id = 201, name ="Television",price = 47909, brand = "LG"},
            new Product() { Id = 301, name ="DvD player", price = 3987, brand = "Onida"},
            new Product() { Id = 401, name ="Refrigrator",price = 28890, brand = "Samsung"},
            new Product() { Id = 501, name ="Laptop", price = 43998, brand = "Dell"}
        };



        //Using For Loop
        for (int i = 0; i < products.Count; i++)
        {
            if (products[i].price > 500)
                Console.WriteLine($"name={products[i].name}, brand={products[i].brand}");
        }

        //Using foreach loop
        foreach (var d in products)
        {
            if (d.price > 500)
                Console.WriteLine($"name={d.name}, brand={d.brand}");
        }
        //using Lambda Expression
        products.Where(x => x.price > 500).ToList().ForEach(x => Console.WriteLine($"name={x.name},
brand={x.brand}"));



        //Even numbers using LINQ Query
        var result = from v in products
                     where v.price > 500
                     select v;
        result.ToList().ForEach(x => Console.WriteLine($"name={x.name}, brand={x.brand}"));

        Console.ReadLine();

    }
  }
}
```

Result:

```
name=Air Conditioner, brand=Daikin
name=Television, brand=LG
name=Air Conditioner, brand=Daikin
name=Television, brand=LG
name=Air Conditioner, brand=Daikin
name=Television, brand=LG
name=Air Conditioner, brand=Daikin
name=Television, brand=LG
```

5. Create your own class and variables and initialize with some valuesforforeachlambdalinq query

Code:

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

namespace Day8Project5
{


        //Author: Triveni Anumolu
        //Program: Own class with own variables and printing by using for,foreach,Lamda,LINQ
    class Office
    {
        public string name;
        public string location;
        public string type;
    }
    class Program
    {
        static void Main(string[] args)
        {
            List<Office> off = new List<Office>()
            {
                new Office() { name = "nbhealthcare", location = "Madhapur", type = "IT" },
                new Office() { name = "eenadu", location = "Irrummanzil", type = "Paper" },
                new Office() { name = "byjus", location = "jubliee hills", type = "education" },
                new Office() { name = "sutherland", location = "Lanco hills", type = "IT" },
                new Office() { name = "accenture", location = "Mindspace", type = "IT" }
```

```
            };
        //for loop
        for (int i = 0; i < off.Count; i++)
        {
            Console.WriteLine($"name={off[i].name}, location={off[i].location}, type={off[i].type}");
        }

        //foreach loop
        foreach (var o in off)
        {
            Console.WriteLine($"name={o.name}, location={o.location}, type={o.type}");
        }

        //Lamda Expression

        off.ToList().ForEach(o => Console.WriteLine($"name={o.name}, location={o.location},
type={o.type}"));

        //LINQ Query
        var result = from o in off
                     select o;
        result.ToList().ForEach(o => Console.WriteLine($"name={o.name}, location={o.location},
type={o.type}"));

        Console.ReadLine();



        }
    }
}
```

Result:

```
name=nbhealthcare, location=Madhapur, type=IT
name=eenadu, location=Irrummanzil, type=Paper
name=byjus, location=jubliee hills, type=education
name=sutherland, location=Lanco hills, type=IT
name=accenture, location=Mindspace, type=IT
name=nbhealthcare, location=Madhapur, type=IT
name=eenadu, location=Irrummanzil, type=Paper
name=byjus, location=jubliee hills, type=education
name=sutherland, location=Lanco hills, type=IT
name=accenture, location=Mindspace, type=IT
name=nbhealthcare, location=Madhapur, type=IT
name=eenadu, location=Irrummanzil, type=Paper
name=byjus, location=jubliee hills, type=education
name=sutherland, location=Lanco hills, type=IT
name=accenture, location=Mindspace, type=IT
name=nbhealthcare, location=Madhapur, type=IT
name=eenadu, location=Irrummanzil, type=Paper
name=byjus, location=jubliee hills, type=education
name=sutherland, location=Lanco hills, type=IT
name=accenture, location=Mindspace, type=IT
```

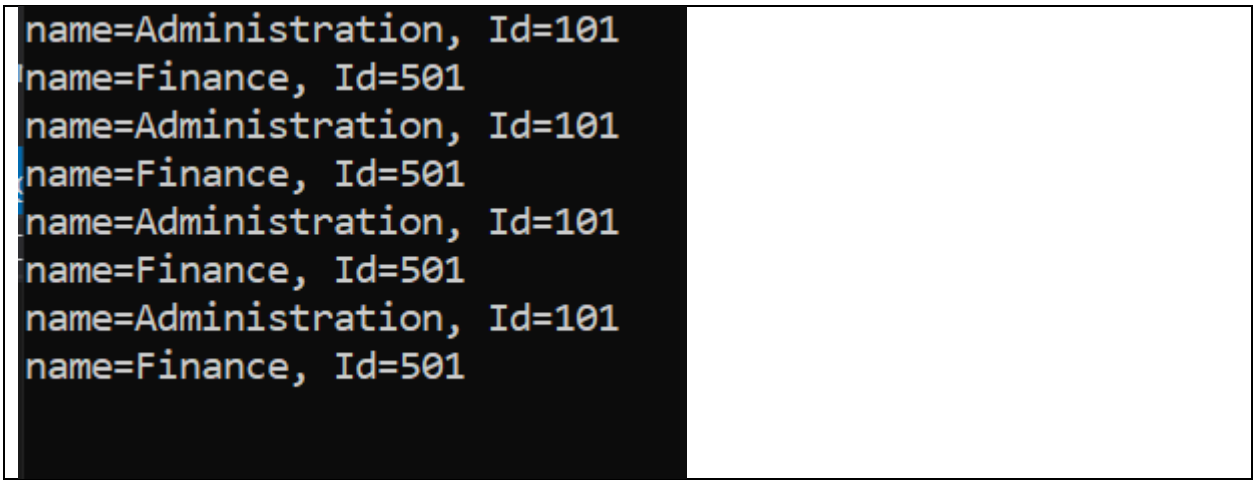
4. Create a Department class and add variables id,name,empcount write code to print id,name of departments whose empcount is greater than 50usingforforeachlambdalinq query

Code:

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

namespace Day8Project4
{
   class Department
   {
      public int Id;
      public string name;
      public int empcount;
   }
   class Program
   {
```

```
      static void Main(string[] args)
      {
         /*****************************************************************
          Author: Triveni Anumolu
         Purpose: Creating a department class and adding variables and printing dept id, dept name
         *****************************************************************/
         List<Department> dept = new List<Department>()
         {
            new Department() { Id = 101, name ="Administration", empcount = 100},

            new Department() { Id = 201, name ="Production", empcount = 50},
            new Department() { Id = 301, name ="Inspection",empcount = 30},
            new Department() { Id = 401, name ="Manufacturing", empcount = 90},
            new Department() { Id = 501, name ="Finance",  empcount = 70}
         };



         //Using For Loop
         for (int i = 0; i < dept.Count; i++)
         {
            if (dept[i].empcount > 50)
               Console.WriteLine($"name={dept[i].name}, Id={dept[i].Id}");
         }

         //Using foreach loop
         foreach (var d in dept)
         {
            if (d.empcount  > 50)
               Console.WriteLine($"name={d.name}, Id={d.Id}");
         }
         //using Lambda Expression
         dept.Where(x => x.empcount > 50).ToList().ForEach(x => Console.WriteLine($"name={x.name},
Id={x.Id}"));



         //Even numbers using LINQ Query
         var result = from v in dept
                  where v.empcount >50
                  select v;
         result.ToList().ForEach(x => Console.WriteLine($"name={x.name}, Id={x.Id}"));

         Console.ReadLine();

      }
   }
}
```

Result:


name=Administration, Id=101
name=Finance, Id=501
name=Administration, Id=101
name=Finance, Id=501
name=Administration, Id=101
name=Finance, Id=501
name=Administration, Id=101
name=Finance, Id=501