| **DAY 8 : Morning Assignment**  **By**  **Vihar D.** |
| --- |

| **Assignment 1** |
| --- |
| **Create a list with 8 values & find even numbers from the list using for, foreach, lambda and LINQ loop types.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace values8\_even\_4looptypes  {  internal class Program  {  static void Main(string[] args)  {  List<int> data = new List<int>() { 65,89,41,78,16,84,39,67,22,64 };    *//using for loop------------------------------------------------------------*  Console.WriteLine("\n Output (using for loop ) : ------------\n");  for (int i = 0; i < data.Count; i++)  {  if (data[i] % 2 == 0)  Console.WriteLine("\t{0}", data[i]);  }  *//using foreach loop------------------------------------------------------------*  Console.WriteLine("\n Output (using foreach loop ) : ------------\n");  foreach (var d in data)  {  if (d % 2 == 0)  Console.WriteLine("\t{0}", d);  }  *//using lambda expression------------------------------------------------------------*  Console.WriteLine("\n Output (using lambda exp ) : ------------\n");  data.Where(x => x % 2 == 0).ToList().ForEach(x => Console.WriteLine("\t{0}", x));  *//using LINQ query------------------------------------------------------------*  Console.WriteLine("\n Output (using LINQ ) : ------------\n");  var output = from d in data  where d % 2 == 0  select d;  output.ToList().ForEach(x => Console.WriteLine("\t{0}", x));  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 2** |
| --- |
| **Create a class of list employees and print using for, foreach, lambda and LINQ loop types.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace class\_emp\_4looptypes  {  class Employee  {  public int emp\_id;  public string emp\_name;  public int emp\_salary;  }  internal class Program  {  static void Main(string[] args)  {  List<Employee> emp = new List<Employee>()  {  new Employee(){emp\_id = 1, emp\_name = "Vihar Dasari", emp\_salary = 50000},  new Employee(){emp\_id = 2, emp\_name = "Sarath Phani", emp\_salary = 45000},  new Employee(){emp\_id = 3, emp\_name = "Manoj Karnatapu", emp\_salary = 35000},  new Employee(){emp\_id = 4, emp\_name = "Manoj Yekkola", emp\_salary = 75000},  new Employee(){emp\_id = 5, emp\_name = "Pavan Chirra", emp\_salary = 60000},  };  *//using for Loop--------------------------------------------------------------*  Console.WriteLine("\n Output (using for loop ) : ------------\n");  for (int i = 0; i < emp.Count; i++)  {  Console.WriteLine($"Employee id = {emp[i].emp\_id}, " +  $"Employee name = {emp[i].emp\_name}, " +  $"Employee salary = {emp[i].emp\_salary}");  }  *//using foreach Loop--------------------------------------------------------------*  Console.WriteLine("\n Output (using foreach loop ) : ------------\n");  foreach (var e in emp)  {  Console.WriteLine($"Employee id = {e.emp\_id}, " +  $"Employee name = {e.emp\_name}, " +  $"Employee salary = {e.emp\_salary}");  }  *//using lambda expression--------------------------------------------------------------*  Console.WriteLine("\n Output (using lambda exp ) : ------------\n");  emp.ForEach(e => Console.WriteLine($"Employee id = {e.emp\_id}, " +  $"Employee name = {e.emp\_name}, " +  $"Employees salary = {e.emp\_salary}"));  *//using LINQ query--------------------------------------------------------------*  Console.WriteLine("\n Output (using LINQ ) : ------------\n");  var output = from e in emp  select e;  output.ToList().ForEach(e => Console.WriteLine($"Employee id = {e.emp\_id}, " +  $"Employee name = {e.emp\_name}, " +  $"Employee salary = {e.emp\_salary}"));  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 3** |
| --- |
| **Create a class of list Product and add variables, then print Product ( name and brand ) whose price is > 500 using for, foreach, lambda and LINQ loop types.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace class\_prod\_4looptypes  {  class Product  {  public int prod\_id;  public string prod\_name;  public int prod\_price;  public string prod\_brand;  }  internal class Program  {  static void Main(string[] args)  {  List<Product> prod = new List<Product>()  {  new Product() { prod\_id = 1, prod\_name = "ROG Strix", prod\_price = 820, prod\_brand = "ASUS"},  new Product() { prod\_id = 2, prod\_name = "Legion 5i", prod\_price = 630, prod\_brand = "Lenovo"},  new Product() { prod\_id = 3, prod\_name = "Pavilion", prod\_price= 250, prod\_brand ="HP"},  new Product() { prod\_id = 4, prod\_name = "Nitro Predator", prod\_price = 350, prod\_brand = "Acer"}  };  *//using for Loop----------------------------------------------------------------------------------------------------------------------------*  Console.WriteLine("\n Output ( using for loop ) : ------------\n");  for (int i = 0; i < prod.Count; i++)  {  if (prod[i].prod\_price > 500)  {  Console.WriteLine($"Product name = {prod[i].prod\_name}, " +  $"Product brand = {prod[i].prod\_brand}");  }  }  *//using foreach Loop----------------------------------------------------------------------------------------------------------------------------*  Console.WriteLine("\n Output ( using foreach loop ) : ------------\n");  foreach (var p in prod)  {  if (p.prod\_price > 500)  Console.WriteLine($"Product name = {p.prod\_name}, " +  $"Product brand ={p.prod\_brand}");  }  *//using lambda expression----------------------------------------------------------------------------------------------------------------------------*  Console.WriteLine("\n Output ( using lambda exp ) : ------------\n");  prod.Where(p => p.prod\_price > 500).ToList().ForEach(p => Console.WriteLine($"Product name = {p.prod\_name}, " +  $"Product brand = {p.prod\_brand}"));  *//using LINQ query----------------------------------------------------------------------------------------------------------------------------*  Console.WriteLine("\n Output ( using LINQ ) : ------------\n");  var output = from p in prod  where p.prod\_price > 500  select p;  output.ToList().ForEach(p => Console.WriteLine($"Product name = {p.prod\_name}, " +  $"Product brand = {p.prod\_brand}"));  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 4** |
| --- |
| **Create a Department class and add variables, then print id & name of departments whose employee count > 50 using for, foreach, lambda and LINQ loop types.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace dept\_4looptypes  {  class Department  {  public int dept\_id;  public string dept\_name;  public int dept\_empCount;  }  internal class Program  {  static void Main(string[] args)  {  List<Department> dept = new List<Department>()  {  new Department() {dept\_id = 1, dept\_name = "Manager", dept\_empCount = 10},  new Department() {dept\_id = 2, dept\_name = "Administration", dept\_empCount = 50},  new Department() {dept\_id = 3, dept\_name = "Logistics", dept\_empCount = 200},  new Department() {dept\_id = 4, dept\_name = "Packaging", dept\_empCount = 350}  };  *//using for Loop----------------------------------------------------------------------------------------------------------------------------*  Console.WriteLine("\n Output ( using for loop ) : ------------\n");  for (int i = 0; i < dept.Count; i++)  {  if (dept[i].dept\_empCount > 50)  {  Console.WriteLine($"Department id = {dept[i].dept\_id}, " +  $"Department name = {dept[i].dept\_name}");  }  }  *//using foreach Loop----------------------------------------------------------------------------------------------------------------------------*  Console.WriteLine("\n Output ( using foreach loop ) : ------------\n");  foreach (var d in dept)  {  if (d.dept\_empCount > 50)  {  Console.WriteLine($"Department id = {d.dept\_id}, " +  $"Department name = {d.dept\_name}");  }  }  *//using lambda expression----------------------------------------------------------------------------------------------------------------------------*  Console.WriteLine("\n Output ( using lambda exp ) : ------------\n");  dept.Where(d => d.dept\_empCount > 50).ToList().ForEach(d => Console.WriteLine($"Department id = {d.dept\_id}, " +  $"Department name = {d.dept\_name}"));  *//using LINQ query----------------------------------------------------------------------------------------------------------------------------*  Console.WriteLine("\n Output ( using LINQ ) : ------------\n");  var output = from d in dept  where d.dept\_empCount > 50  select d;  output.ToList().ForEach(d => Console.WriteLine($"Department id = {d.dept\_id}, " +  $"Department name = {d.dept\_name}"));  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 5** |
| --- |
| **Pictorially represent class and multiple objects** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace random\_class\_4looptypes  {  class bikes  {  public string bike\_model;  public string bike\_brand;  public int bike\_make\_year;  public int bike\_powercc;  }  internal class Program  {  static void Main(string[] args)  {  List<bikes> bike = new List<bikes>()  {  new bikes(){ bike\_model = "Svartpilen 401", bike\_brand = "Husqvarna", bike\_make\_year = 2015, bike\_powercc = 375},  new bikes(){ bike\_model = "Duke 250", bike\_brand = "KTM", bike\_make\_year = 2012, bike\_powercc = 250},  new bikes(){ bike\_model = "ThunderBird 350x", bike\_brand = "Royal Enfield", bike\_make\_year = 2018, bike\_powercc = 350},  new bikes(){ bike\_model = "GS750r", bike\_brand = "BMW", bike\_make\_year = 2016, bike\_powercc = 750},  new bikes(){ bike\_model = "Panigale 899", bike\_brand = "Ducati", bike\_make\_year = 2019, bike\_powercc = 900},  new bikes(){ bike\_model = "CB500x", bike\_brand = "Honda", bike\_make\_year = 2021, bike\_powercc = 500}  };  *//all bike models----------------------------------------------------------------*  Console.WriteLine("\n------------ All Bike Models ( using for loop ) : ------------\n");  for (int i = 0; i < bike.Count; i++)  {  Console.WriteLine($"Bike model = {bike[i].bike\_model}, " +  $"Bike brand = {bike[i].bike\_brand}, " +  $"Bike make year = {bike[i].bike\_make\_year}, " +  $"Bike CC = {bike[i].bike\_powercc}");  }  *//using for loop----------------------------------------------------------------*  Console.WriteLine("\n------------ Bike Models >= 500cc ( using for loop ) : ------------\n");  for (int i = 0; i < bike.Count; i++)  {  if (bike[i].bike\_powercc >= 500)  {  Console.WriteLine($"Bike model = {bike[i].bike\_model}, " +  $"Bike brand = {bike[i].bike\_brand}, " +  $"Bike make year = {bike[i].bike\_make\_year}, " +  $"Bike CC = {bike[i].bike\_powercc}");  }  }  *//using foreach loop----------------------------------------------------------------*  Console.WriteLine("\n------------ Bike Models >= 500cc ( using foreach loop ) : ------------\n");  foreach (var b in bike)  {  if(b.bike\_powercc >= 500)  {  Console.WriteLine($"Bike model = {b.bike\_model}, " +  $"Bike brand = {b.bike\_brand}, " +  $"Bike make year = {b.bike\_make\_year}, " +  $"Bike CC = {b.bike\_powercc}");  }  }  *//using lambda expression----------------------------------------------------------------*  Console.WriteLine("\n------------ Bike Models >= 500cc ( using lambda exp ) : ------------\n");  bike.Where(b => b.bike\_powercc >= 500).ToList().ForEach(b => Console.WriteLine($"Bike model = {b.bike\_model}, " +  $"Bike brand = {b.bike\_brand}, " +  $"Bike make year = {b.bike\_make\_year}, " +  $"Bike CC = {b.bike\_powercc}"));  *//using LINQ query----------------------------------------------------------------*  Console.WriteLine("\n------------ Bike Models >= 500cc ( using LINQ ) : ------------\n");  var output = from b in bike  where b.bike\_powercc >= 500  select b;  output.ToList().ForEach(b => Console.WriteLine($"Bike model = {b.bike\_model}, " +  $"Bike brand = {b.bike\_brand}, " +  $"Bike make year = {b.bike\_make\_year}, " +  $"Bike CC = {b.bike\_powercc}"));  Console.ReadLine();  }  }  } |
| **Output :** |
|  |