| **DAY 7 : Morning Assignment**  **By**  **Vihar D.** |
| --- |

| **Assignment 1** |
| --- |
| **Create an Employee class with 3 variables and 2 methods ReadEmployee() and PrintEmployee() 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 \_3var2meth\_read\_print  {  class Employee  {  public int id;  public string name;  public int salary;  public void ReadEmp()  {  Console.WriteLine("Enter id :");  id = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter name :");  name = Console.ReadLine();    Console.WriteLine("Enter salary :");  salary = Convert.ToInt32(Console.ReadLine());  }  public void PrintEmp()  {  Console.WriteLine($"id={id},name={name},salary={salary}");  *//Console.WriteLine("id={0},name={1},salary={2}", id, name, salary);*  *//Console.WriteLine("id =" + id + ", Name =" + name + ", salary =" + salary);*  }  }  internal class Program  {  static void Main(string[] args)  {  Employee emp1 = new Employee();  emp1.ReadEmp();  emp1.PrintEmp();  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 2** |
| --- |
| **Write the 3 definitions of class and 4 points about objects discussed in Day 7 morning session.** |
| **Answer :** |
| **Class :**   * It’s a collection of variables and methods. * It's a blueprint to create objects as per requirements. * It consists of a State and Behavior where State is basically **variables** and Behavior deals with the methods inside the given class.   **Object :**   * An object is an instance of a class. * Any number of objects can be created. * Memory is occupied by objects when they are created. * Objects are of reference types in general. |
| **Output :** |
|  |

| **Assignment 3** |
| --- |
| **Pictorially represent class and multiple objects** |
| **Answer :** |
|  |
| **Output :** |
|  |

| **Assignment 4** |
| --- |
| **Create the given classes : 1.Customer 2.Product 3.Seller 4.Department** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace \_4classes  {  internal class Program  {  static void Main(string[] args)  {  *//Customer---------------------------------------------*  customer cust1 = new customer();  Console.WriteLine("Enter Customer Details : ");  cust1.CreateCustData();  *//Product----------------------------------------------*  product prod1 = new product();  Console.WriteLine("Enter Product Details : ");  prod1.CreateProdData();  *//Seller-----------------------------------------------*  seller sell1 = new seller();  Console.WriteLine("Enter Seller Details : ");  sell1.CreateSellData();  *//Department-------------------------------------------*  department dept1 = new department();  Console.WriteLine("Enter Department Details : ");  dept1.CreateDeptData();  cust1.DisplayCustData();  prod1.DisplayProdData();  sell1.DisplaySellData();  dept1.DisplayDeptData();    Console.WriteLine("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* EXECUTION TERMINATED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 5** |
| --- |
| **Create an Employee class with 3 public variables. Create an object and initialize with values while creating an object and print the values.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace emp\_public  {  class employee  {  public int emp\_id;  public string emp\_name;  public int emp\_age;  public int emp\_salary;  }  internal class Program  {  static void Main(string[] args)  {  employee emp = new employee()  {  emp\_id = 5000,  emp\_name = "Vihar Dasari",  emp\_age = 23,  emp\_salary = 10000  };  Console.WriteLine($"\n Employee id = {emp.emp\_id}" +  $"\n Employee name = {emp.emp\_name}" +  $"\n Employee age = {emp.emp\_age}" +  $"\n Employee salary = {emp.emp\_salary}" );  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 6** |
| --- |
| **Create an Employee class with its array object and initialize with 5 employees. Print output using for, foreach and lambda expression loop types.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace emp5\_3looptypes  {  internal class Program  {  class employee  {  public int emp\_id;  public string emp\_name;  public int emp\_salary;  }  static void Main(string[] args)  {  employee[] emp = new 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},  new employee(){emp\_id = 6, emp\_name = "Vamsi Krishna", emp\_salary = 40000},  };  *//using for loop--------------------------------------------------*  Console.WriteLine("\n Output (using for loop) : ");  for (int i=0;i<emp.Length;i++)  {  Console.WriteLine($"Employee id = {emp[i].emp\_id}, " +  $"Employee name = {emp[i].emp\_name}, " +  $"Employee salary = {emp[i].emp\_salary}");  }  *//using foreach loop-----------------------------------------------*  Console.WriteLine("\n Output (using foreach loop) : ");  foreach(var e in emp)  {  Console.WriteLine($"Employee id = {e.emp\_id}, " +  $"Employee name = {e.emp\_name}, " +  $"Employee salary = {e.emp\_salary}");  }  *//using Lambda Expression*  Console.WriteLine("\n Output (using Lambda exp) : ");  emp.ToList().ForEach(d => Console.WriteLine($"Employee id = {d.emp\_id}, " +  $"Employee name = {d.emp\_name}, " +  $"Employee salary = {d.emp\_salary}"));  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 7** |
| --- |
| **Similar to Assignment 6, write a C# code to print employees whose salary is >=50000 using for, foreach and lambda expression loop types.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace emp5\_3looptypes\_gt  {  internal class Program  {  class employee  {  public int emp\_id;  public string emp\_name;  public int emp\_salary;  }  static void Main(string[] args)  {  employee[] emp = new 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 (>= 50,000) (using for loop) :\n");  for (int i = 0; i < emp.Length; i++)  {  if (emp[i].emp\_salary >= 50000)  Console.WriteLine($"Employee id = {emp[i].emp\_id}, " +  $"Employee name = {emp[i].emp\_name}, " +  $"Employee salary = {emp[i].emp\_salary}");  }  *//using foreach loop----------------------------------------------------------------------------------------------*  Console.WriteLine("\n Output (>= 50,000) (using foreach loop) :\n");  foreach (var e in emp)  {  if (e.emp\_salary >= 50000)  Console.WriteLine($"Employee id = {e.emp\_id}, " +  $"Employee name = {e.emp\_name}, " +  $"Employee salary = {e.emp\_salary}");  }  *//Using Lambda Expression----------------------------------------------------------------------------------------------*  Console.WriteLine("\n Output (>= 50,000) (using Lambda exp) :\n");    emp.ToList().Where(e => e.emp\_salary >= 50000).ToList().ForEach(e =>Console.WriteLine($"Employee id = {e.emp\_id}," +  $"Employee name = {e.emp\_name}," +  $"Employee salary = {e.emp\_salary}"));  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 8** |
| --- |
| **Similar to Assignment 6 & 7, Create a list of Customer and Product arrays using for, foreach and lambda expression loop types.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace cust\_prod\_3looptypes  {  class customer  {  public int cust\_id;  public string cust\_name;  public string cust\_subtype;  }  class product  {  public int prod\_id;  public string prod\_name;  public int prod\_price;  }  internal class Program  {  static void Main(string[] args)  {  *//Customers Class----------------------------------------------------------------------*  customer[] cust = new customer[]  {  new customer(){ cust\_id = 1, cust\_name = "Vihar Dasari", cust\_subtype = "Prime"},  new customer(){ cust\_id = 2, cust\_name = "Pavan Chirra", cust\_subtype = "General"},  new customer(){ cust\_id = 3, cust\_name = "Manoj Karnatapu", cust\_subtype = "Prime"}  };  *//using for loop-----------------------------------------------------------------------*  Console.WriteLine("\n Output (using for loop) : ");  for (int i = 0; i < cust.Length; i++)  {  Console.WriteLine($"Customer id = {cust[i].cust\_id}, " +  $"Customer name = {cust[i].cust\_name}, " +  $"Customer subtype = {cust[i].cust\_subtype}");  }  *//using foreach loop-----------------------------------------------------------------------*  Console.WriteLine("\n Output (using foreach loop) : ");  foreach (var e in cust)  {  Console.WriteLine($"Customer id = {e.cust\_id}, " +  $"Customer name = {e.cust\_name}, " +  $"Customer subtype = {e.cust\_subtype}");  }  *//using Lambda Expression-----------------------------------------------------------------------*  Console.WriteLine("\n Output (using lambda exp) : ");  cust.ToList().ForEach(d => Console.WriteLine($"Customer id = {d.cust\_id}, " +  $"Customer name = {d.cust\_name}, " +  $"Customer subtype = {d.cust\_subtype}"));  Console.WriteLine("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  *//Products Class----------------------------------------------------------------------*  product[] prod = new product[]  {  new product(){ prod\_id = 10, prod\_name = "Asus ROG", prod\_price = 80000},  new product(){ prod\_id = 20, prod\_name = "MSI Gaming", prod\_price = 75000},  new product(){ prod\_id = 30, prod\_name = "Gigabyte Gaming", prod\_price = 50000}  };  *//using for loop-----------------------------------------------------------------------*  Console.WriteLine("\n Output (using for loop) : ");  for (int i = 0; i < prod.Length; i++)  {  Console.WriteLine($"Product id = {prod[i].prod\_id}, " +  $"Product name = {prod[i].prod\_name}, " +  $"Product price = {prod[i].prod\_price}");  }  *//using foreach loop-----------------------------------------------------------------------*  Console.WriteLine("\n Output (using foreach loop) : ");  foreach (var p in prod)  {  Console.WriteLine($"Product id = {p.prod\_id}, " +  $"Product name = {p.prod\_name}, " +  $"Product price = {p.prod\_price}");  }  *//using Lambda Expression-----------------------------------------------------------------------*  Console.WriteLine("\n Output (using lambda exp) : ");  prod.ToList().ForEach(d => Console.WriteLine($"Product id = {d.prod\_id}, " +  $"Product name = {d.prod\_name}, " +  $"Product price = {d.prod\_price}"));  Console.ReadLine();  }  }  } |
| **Output :** |
|  |