DAY 7 ASSIGNMENT

by SANJAY .K

|  |
| --- |
| 1. Create Employee class with three variables and two 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 Day\_\_7\_mng1st\_feb2022\_project\_1  {  class Employee  {  public int id;  public string name;  public int salary;  public void ReadEmployee()  {  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 PrintEmployee()  {  Console.WriteLine($"Id={id},Name={name},Salary={salary}");  }  }  internal class Program  {  static void Main(string[] args)  {  Employee emp1 = new Employee();  emp1.ReadEmployee();  emp1.PrintEmployee();  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| 2. Write the 3 def of class and 4 points about object discussed in the class. |
| Class |
| * Class is group of variables and methods. * A class consist of state and bhaviour. * A class is like a design to create objects. |
| Object |
| * An object is an instance of a class * We can create any num of objects * Objects occupy memory * Objects are reference type |

|  |
| --- |
| 3. Pictorially represent class and multiple objects |
| pic day7.png |

|  |
| --- |
| 4. Create below classes:  1. Customer  2. Product  3. Seller  4. Department |
| 1. Customer Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day\_\_7\_mng1st\_feb2022\_project\_1  {  class Customer  {  public int id;  public string name;  public long phnno;  public void ReadCustomer()  {  Console.WriteLine("Enter id:");  id=Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter name:");  name = Console.ReadLine();  Console.WriteLine("Enter phnno:");  phnno = Convert.ToInt64(Console.ReadLine());  }  public void PrintCustomer()  {  Console.WriteLine($"Id={id},Name={name},Phnno={phnno}");  }  }  internal class Program  {  static void Main(string[] args)  {  Customer customer1 = new Customer();  customer1.ReadCustomer();  customer1.PrintCustomer();  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| 1. Product Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day\_\_7\_mng1st\_feb2022\_project\_1  {  class Product  {  public int id;  public string name;  public int price;  public void ReadProduct()  {  Console.WriteLine("Enter id:");  id=Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter product name:");  name = Console.ReadLine();  Console.WriteLine("Enter product price:");  price = Convert.ToInt32(Console.ReadLine());  }  public void PrintProduct()  {  Console.WriteLine($"Id={id},Name={name},Price={price}");  }  }  internal class Program  {  static void Main(string[] args)  {  Product product1 = new Product();  product1.ReadProduct();  product1.PrintProduct();  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| 3.Seller Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day\_\_7\_mng1st\_feb2022\_project\_1  {  class Seller  {  public int id;  public string name;  public string place;  public void ReadSeller()  {  Console.WriteLine("Enter id:");  id=Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter seller name:");  name = Console.ReadLine();  Console.WriteLine("Enter seller place:");  place = Console.ReadLine();  }  public void PrintSeller()  {  Console.WriteLine($"Id={id},Name={name},Place={place}");  }  }  internal class Program  {  static void Main(string[] args)  {  Seller seller1 = new Seller();  seller1.ReadSeller();  seller1.PrintSeller();  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| 4.Department |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day\_\_7\_mng1st\_feb2022\_project\_1  {  class Department  {  public int id;  public string name;  public string place;  public void ReadDepartment()  {  Console.WriteLine("Enter id:");  id=Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter department name:");  name = Console.ReadLine();  Console.WriteLine("Enter department place:");  place = Console.ReadLine();  }  public void PrintDepartment()  {  Console.WriteLine($"Id={id},Name={name},Place={place}");  }  }  internal class Program  {  static void Main(string[] args)  {  Department department1 = new Department();  department1.ReadDepartment();  department1.PrintDepartment();  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| 5. Create Employee class with 3 public variables.  Create Employee object and initialize with values while creating object  and print the values. |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day\_7\_1stfeb\_2022\_mng\_project\_2  {  class Employee  {  public int id;  public string name;  public int salary;  }  internal class Program  {  static void Main(string[] args)  {  Employee emp = new Employee() { id = 1, name = "John", salary = 5000 };  Console.WriteLine($"id={emp.id},name={emp.name},Salary={emp.salary}");  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| 6. now create employees array object and initialize with 5 employees  write code using  a. for loop  b. foreach loop  c. lambda expression. |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day\_7\_1stfeb\_2022\_mng\_project\_2  {  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 = 1, name ="john", salary =30000},  new Employee() { id = 2, name ="prasad",salary=25000},  new Employee() { id = 3, name ="sai",salary=10000},  new Employee() { id = 4, name ="uday",salary=64000},  new Employee() { id = 5, name ="jp",salary=40000}  };  //using for loop  for(int i = 0; i<employees.Length; i++)  {  Console.WriteLine($"id={employees[i].id},name={employees[i].id},salary={employees[i].id}");  }  //using foreach  foreach(var e in employees)  {  Console.WriteLine($"id={e.id},name={e.name},salary{e.salary}");  }  //using lamda expression  employees.ToList().ForEach(e => Console.WriteLine($"id={e.id},name={e.name},salary{e.salary}"));  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| 7. For the above project,  write code to print employees who is getting salary >=5000 using  for loop  foreach loop  lambda expression |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day\_7\_1stfeb\_2022\_mng\_project\_2  {  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 = 1, name ="john", salary =30000},  new Employee() { id = 2, name ="prasad",salary=25000},  new Employee() { id = 3, name ="sai",salary=10000},  new Employee() { id = 4, name ="uday",salary=64000},  new Employee() { id = 5, name ="jp",salary=40000}  };  //using for loop  for(int i = 0; i<employees.Length; i++)  {  if(employees[i].salary>=30000)  Console.WriteLine($"id={employees[i].id},name={employees[i].name},salary={employees[i].salary}");  }  //using foreach  foreach(var e in employees)  {  if(e.salary>=30000)  Console.WriteLine($"id={e.id},name={e.name},salary{e.salary}");  }  //using lamda expression  employees.ToList().Where(e => e.salary>=30000).ToList().ForEach(e => Console.WriteLine($"id={e.id},name={e.name},salary{e.salary}"));  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| 8. Similar to 6 and 7 projects create list of Customer an Product Arrays  and practice for, foreach and lambda expression |
| Customer Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day\_7\_1stfeb\_2022\_mng\_project\_2  {  class Customer  {  public int id;  public string name;  public int points;  }  internal class Program  {  static void Main(string[] args)  {  Customer[] customer = new Customer[]  {  new Customer() { id = 1, name ="john", points =95},  new Customer() { id = 2, name ="prasad",points=25},  new Customer() { id = 3, name ="sai",points=10},  new Customer() { id = 4, name ="uday",points=64},  new Customer() { id = 5, name ="jp",points=40}  };  //using for loop  for (int i = 0; i < customer.Length; i++)  {  if (customer[i].points >= 30)  Console.WriteLine($"id={customer[i].id},name={customer[i].name},Points={customer[i].points}");  }  //using foreach  foreach (var e in customer)  {  if (e.points >= 30)  Console.WriteLine($"id={e.id},name={e.name},Points{e.points}");  }  //using lamda expression  customer.ToList().Where(e => e.points >= 30).ToList().ForEach(e => Console.WriteLine($"id={e.id},name={e.name},Points{e.points}"));  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| Product Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day\_7\_1stfeb\_2022\_mng\_project\_2  {  class Product  {  public int id;  public string name;  public int price;  }  internal class Program  {  static void Main(string[] args)  {  Product[] product = new Product[]  {  new Product() { id = 1, name ="john", price =955},  new Product() { id = 2, name ="prasad",price=255},  new Product() { id = 3, name ="sai",price=10},  new Product() { id = 4, name ="uday",price=64},  new Product() { id = 5, name ="jp",price=40}  };  //using for loop  for (int i = 0; i < product.Length; i++)  {  if (product[i].price >= 100)  Console.WriteLine($"id={product[i].id},name={product[i].name},Price={product[i].price}");  }  //using foreach  foreach (var e in product)  {  if (e.price >= 100)  Console.WriteLine($"id={e.id},name={e.name},Price{e.price}");  }  //using lamda expression  product.ToList().Where(e => e.price >= 100).ToList().ForEach(e => Console.WriteLine($"id={e.id},name={e.name},Price{e.price}"));  Console.ReadLine();  }  }  } |
| Output |
|  |