1) Write the 3 def of class and 4 points about object discussed in the class.

A)Class:

a)It is like a blueprint which has all details to create an object

b)It has variables and methods

c)It has states and behaviours

B)Object:

a)It is an instance of a class

b)A class can have multiple objects

c)A class can have objects of different sizes depending on their values of variables

d)Objects are of reference data type

|  |
| --- |
| . 2)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 D7Project1  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : Create Employee class with three variables and two methods  ReadEmployee and PrintEmployee and create an object and call methods.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  class Employee  {  public int employeeID;  public string employeeName;  public int employeeSalary;  public void ReadEmployee()  {  Console.Write("\nEnter employee ID : ");  employeeID = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter employee name : ");  employeeName = (Console.ReadLine());  Console.Write("Enter employee salary : ");  employeeSalary = Convert.ToInt32(Console.ReadLine());  }  public void PrintEmployee()  {  Console.Write("\nEmployee ID : " + employeeID);  Console.Write("\nEmployee name: " + employeeName);  Console.Write("\nEmployee salary : " + employeeSalary);  }  }  internal class Program  {  static void Main(string[] args)  {  Employee emp1 = new Employee();  Employee emp2 = new Employee();    emp1.ReadEmployee();  Console.Write("\n");  emp2.ReadEmployee();  Console.Write("\n\n");  emp1.PrintEmployee();  Console.Write("\n");  emp2.PrintEmployee();  Console.ReadLine();  }    }  } |
| OUTPUT |
|  |

|  |
| --- |
| 3. Create below classes:  a) Customer  b)Product  c) Seller  d) Department |
| CODE |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace D7Project2  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : Create below classes:  1. Customer  2. Product  3. Seller  4. Department  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  class Customer  {  public string customerName;  public string customerID;  public string customerAddress;  public long customerContactNumber;  public string customerMailID;  public void AddCustomer()  {  //todo  }  public void DeleteCustomer()  {  //todo  }  public void CustomertActivity()  {  //todo  }  public void DisplayCustomer()  {  //todo  }  public void CustomerTransactions()  {  //todo  }  }  internal class Department  {  public string departmentName;  public string departmentId;  public string departmentHead;  public int departmentFunds;  public int departmentExpenses;  public void AddDepartment()  {  //todo  }  public void DeleteDepartment()  {  //todo  }  public void UpdateDepartment()  {  //todo  }  public void DisplayDepartment()  {  //todo  }  public void DepartmentTransactions()  {  //todo  }  }  internal class Seller  {  public string sellerName;  public string sellerID;  public string sellerAddress;  public long sellerContactNumber;  public string sellerMailID;  public void AddSeller()  {  //todo  }  public void DeleteSeller()  {  //todo  }  public void DisplaySeller()  {  //todo  }  public void SellerTransactions()  {  //todo  }  public void SellerRelativePerformance()  {  //todo  }  }  internal class Product  {  public string productCategory;  public string productBrand;  public string productID;  public string productName;  public int productPrice;  public void AddProduct()  {  //todo  }  public void DeleteProduct()  {  //todo  }  public void DisplayProduct()  {  //todo  }  public void ProductSales()  {  //todo  }  public void ProductRelativePerformance()  {  //todo  }  }  internal class Program  {  static void Main(string[] args)  {  }  }  } |
| OUTPUT |
| NO OUTPUT |

|  |
| --- |
| 4) 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 D7Project3  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose :Create Employee class with 3 public variables.  Create Employee object and initialize with values while  creating object and print the values.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  internal class Employee  {  public int employeeSalary;  public string employeeName;  public int employeeID;  }  internal class Program  {  static void Main(string[] args)  {  Employee emp1 = new Employee() { employeeID = 1,employeeName = "Meena",employeeSalary = 100000};  Console.WriteLine($@"ID : {emp1.employeeID}  Name : {emp1.employeeName}  Salary : {emp1.employeeSalary}");  Console.ReadLine();  }  }  } |
| OUTPUT |
|  |

|  |
| --- |
| 5) 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 Day7Project4  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : Create employees array object and  initialize with 5 employees  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  internal 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 = "Sanjana",salary = 100000 },  new Employee(){id = 2,name = "Akshaya",salary = 150000},  new Employee(){id = 3,name = "Keerthi",salary = 100000},  new Employee(){id = 4,name = "Srusti",salary = 100000},  new Employee(){id = 5,name = "Havya",salary = 200000}  };  //printing values using for loop  for (int i = 0; i < employees.Length; i++)  Console.Write($"\nID : {employees[i].id}\tName : {employees[i].name}\tSalary : {employees[i].salary}");  Console.WriteLine("\n\n");  //printing values using foreach  foreach (var employee in employees)  Console.Write($"\nID : {employee.id}\tName : {employee.name}\tSalary : {employee.salary}");  Console.WriteLine("\n\n");  //printing values using lambda expression  employees.ToList().ForEach(employee => Console.Write($"\nID : {employee.id}\tName : {employee.name}\tSalary : {employee.salary}"));  Console.ReadLine();  }  }  } |
| OUTPUT |
|  |

|  |
| --- |
| 6) For the above project,write code to print employees who is getting salary >=5000 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 Day7Project5  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : For Employee class,write code to print  employees who is getting salary >=5000 using  for loop  foreach loop  lambda expression  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  internal 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 = "Sanjana",salary = 100000 },  new Employee(){id = 2,name = "Akshaya",salary = 150000},  new Employee(){id = 3,name = "Keerthi",salary = 100000},  new Employee(){id = 4,name = "Srusti",salary = 100000},  new Employee(){id = 5,name = "Havya",salary = 200000}  };  //printing employee details with salary>=150000 using for loop  for (int i = 0; i < employees.Length; i++)  {  if (employees[i].salary >= 150000)  Console.Write($"\nID : {employees[i].id}\t\tName : {employees[i].name}\t\tSalary : {employees[i].salary}");  }  Console.WriteLine("\n\n");  //printing employee details with salary>=150000 using foreach loop  foreach(var employee in employees)  {  if (employee.salary >= 150000)  Console.Write($"\nID : {employee.id}\t\tName : {employee.name}\t\tSalary : {employee.salary}");  }  Console.WriteLine("\n\n");  //printing employee details with salary>=150000 using lambda expression  employees.ToList().Where(employee => employee.salary >= 150000).ToList().ForEach(employee => Console.Write($"\nID : {employee.id}\t\tName : {employee.name}\t\tsalary : {employee.salary}"));  Console.ReadLine();  }}} |
| OUTPUT |
|  |

|  |
| --- |
| 7) Create list of Customer and practice for, foreach and lambda expression |
| CODE |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace DayProject6  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author: G V S S SRI LASYA  Purpose :Create customers list and  initialize with 5 customers  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  class Customer  {  public string name;  public int id;  public string mailID;  }  internal class Program  {  static void Main(string[] args)  {  List<Customer> customers = new List<Customer>()  {  new Customer(){id = 1,name = "Sanjana",mailID = "sanjana@abc.com" },  new Customer(){id = 2,name = "Akshaya",mailID = "akshaya@abc.com"},  new Customer(){id = 3,name = "Keerthi",mailID = "keerthi@abc.com"},  new Customer(){id = 4,name = "Srusti",mailID = "srusti@abc.com"},  new Customer(){id = 5,name = "Havya",mailID = "havya@abc.com"}  };  //printing values using for loop  for (int i = 0; i < customers.Count; i++)  Console.Write($"\nID : {customers[i].id}\tName : {customers[i].name}\tMail ID : {customers[i].mailID}");  Console.WriteLine("\n\n");  //printing values using foreach  foreach (var customer in customers)  Console.Write($"\nID : {customer.id}\tName : {customer.name}\tMail ID : {customer.mailID}");  Console.WriteLine("\n\n");  //printing values using lambda expression  customers.ForEach(customer => Console.Write($"\nID : {customer.id}\tName : {customer.name}\tMail ID : {customer.mailID}"));  Console.ReadLine();  }  }  } |
| OUTPUT |
|  |

|  |
| --- |
| 8)Cretae Customers List and initialise with 5 customers and write if condition of id>=3 and print 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 DayProject6  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author: G V S S SRI LASYA  Purpose :Create customers List and  initialize with 5 customers with an if conidtion of id >=3  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  class Customer  {  public string name;  public int id;  public string mailID;  }  internal class Program  {  static void Main(string[] args)  {  List<Customer> customers = new List<Customer>()  {  new Customer(){id = 1,name = "Sanjana",mailID = "sanjana@abc.com" },  new Customer(){id = 2,name = "Akshaya",mailID = "akshaya@abc.com"},  new Customer(){id = 3,name = "Keerthi",mailID = "keerthi@abc.com"},  new Customer(){id = 4,name = "Srusti",mailID = "srusti@abc.com"},  new Customer(){id = 5,name = "Havya",mailID = "havya@abc.com"}  };  //printing values with cutomer id >=3 using for loop  for (int i = 0; i < customers.Count; i++)  {  if (customers[i].id >= 3)  Console.Write($"\nID : {customers[i].id}\tName : {customers[i].name}\tMail ID : {customers[i].mailID}");  }  Console.WriteLine("\n\n");  //printing values with cutomer id >=3 using foreach  foreach (var customer in customers)  {  if (customer.id >= 3)  Console.Write($"\nID : {customer.id}\tName : {customer.name}\tMail ID : {customer.mailID}");  }  Console.WriteLine("\n\n");  //printing values with cutomer id >=3 using lambda expression  customers.ToList().Where(customer => customer.id >= 3).ToList().ForEach(customer => Console.Write($"\nID : {customer.id}\tName : {customer.name}\tMail ID : {customer.mailID}"));  Console.ReadLine();  }  }  } |
| OUTPUT |
|  |

|  |
| --- |
| 9) Create list of Product and practice for, foreach and lambda expression |
| CODE |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace DAY7Project8  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : Create producT list and  initialize with 5 producs  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  class Product  {  public int id;  public string name;  public int price;  internal class Program  {  static void Main(string[] args)  {  List<Product> products = new List<Product>()  {  new Product(){id = 1,name = "product1",price = 5000 },  new Product(){id = 2,name = "product2",price = 5000},  new Product(){id = 3,name = "product3",price = 10000},  new Product(){id = 4,name = "product4",price = 3000},  new Product(){id = 5,name = "product5",price = 4000}  };  //printing values using for loop  for (int i = 0; i < products.Count; i++)  Console.Write($"\nID : {products[i].id}\tName : {products[i].name}\tPrice : {products[i].price}");  Console.WriteLine("\n\n");  //printing values using foreach  foreach (var product in products)  Console.Write($"\nID : {product.id}\tName : {product.name}\tPrice : {product.price}");  Console.WriteLine("\n\n");  //printing values using lambda expression  products.ToList().ForEach(product => Console.Write($"\nID : {product.id}\tName : {product.name}\tPrice : {product.price}"));  Console.ReadLine();  }  }  }  } |
| OUTPUT |
|  |

|  |
| --- |
| 10)Cretae Product List and initialise with 5 product and write if condition of id>=3 and print 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 DAY7Project8  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : Create product list and  initialize with 5 products with an if conidtion of id >=3  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  class Product  {  public int id;  public string name;  public int price;  internal class Program  {  static void Main(string[] args)  {  List<Product> products = new List<Product>()  {  new Product(){id = 1,name = "product1",price = 5000 },  new Product(){id = 2,name = "product2",price = 5000},  new Product(){id = 3,name = "product3",price = 10000},  new Product(){id = 4,name = "product4",price = 3000},  new Product(){id = 5,name = "product5",price = 4000}  };  //printing values with id >=3 using for loop  for (int i = 0; i < products.Count; i++)  {  if (products[i].id >= 3)  Console.Write($"\nID : {products[i].id}\tName : {products[i].name}\tPrice : {products[i].price}");  }  Console.WriteLine("\n\n");  //printing values with id >=3 using foreach  foreach (var product in products)  {  if (product.id >= 3)  Console.Write($"\nID : {product.id}\tName : {product.name}\tPrice : {product.price}");  }  Console.WriteLine("\n\n");  //printing values with id >=3 using lambda expression  products.ToList().Where(product => product.id >=3).ToList().ForEach(product => Console.Write($"\nID : {product.id}\tName : {product.name}\tPrice : {product.price}"));  Console.ReadLine();  }  }  }  } |
| OUTPUT |
|  |