**Capita**

1. **What are Pillars of Oops**

When we say oops (Object Oriented programming ) then we are going to have some real world object .

When we say real world object means , there is some way of creating these objects or Template / or blue print that’s called class

When we are creating any class then it will have some set of definition , these definition could be what type of property Object will have and how it will behave or those property

Behavior of Class will be defined by set of method.

Definition of class will depend { Inheritence , Polymorphism, Abstraction , encapsulation }

e.g. Student…

Encapsulation : Is Binding of DataMember and member function together and hidning complexity of design and expose only which are required .

E.g. Customer { Code, Name } , Database connection , Validation

Inheritance : Inheriting behaviour of parent into child or process of creating new class by extending existing class.

Reusability , private member of parent class is inherited in child class but not accessible in child class.

Type : { single, multilevel , multiple}

Diamond problem : Ambiguity issue

A

C

D

B

public class A

{

public virtual void show()

{

Console.WriteLine("A");

}

}

public class B:A

{

public override void show()

{

Console.WriteLine("B");

}

}

public class C : A

{

public override void show()

{

Console.WriteLine("C");

}

}

public class D: B,C // D cann’t have multiple base class

{

public void print()

{

show(); //which show it will use

show();

Console.WriteLine("D");

}

}

Ambiguty error.

Polymorphism :

Many bevavior of Object and data Mamber

Add(int , int); Add(int , int, double);

1. What is different between Ref and Out ?

In reference we need to initialize reference object , while in case of Out it is not required .

But in out we need to initialize in called method. Otherwise it thow error.

1. Int i=1;

Int j=2;

Funcref(ref j); // 2\*2=4

Funcref(out i)

Funcref(ref int j)

{

J=j\*j;

}

Funcout(out int i)

{

J=j+j // it will throw error because in out we have initialized j

}

1. What is delegate and Event ?

Delegate is typesafe pointer to a function , it hold reference of function and invoke method.

Calling method using delegate is faster compare to calling from object of that class.

Delegate is type safe because signature should be matched with with function signature .

E.g. Employee Table { IBM handing final check }

There are different type of employee like Outworks, WNS and we want to promote those employee , but All class will have first their check list to promote and this definition is not defined in IBMEmployee class but different vendor but final checklist in IBM Employee class.

IBM class have a method which receive List of Employee of Different vendor and condition as delegate defined in vendor application in second parameters.

e.g.

using System;

using System.Collections.Generic;

namespace DelegateEventExpl

{

// Suppose we have Employee class (IBM and there are no of verndor and we want to promote those employee

// but final check is with IBM , so we will not add condition in Employee class to promote employee of each types of vendor

// condition of promotion will depends upon the Vendors and they will pass the list of employee to IBM's mail Employee

// class where will trigger condition and prmote those employee as per differnt vendor 's condition defined in differnt vendor

// class

class Program

{

public delegate bool isPromotionApplicable(Employee e);

public class Employee

{

public int ID { get; set; }

public string Name { get; set; }

public int Salary { get; set; }

public int Exp { get; set; }

public void Promoteemployee(List<Employee> empList , isPromotionApplicable isPromotion)

{

foreach(Employee e in empList)

{

if(isPromotion(e))

{

Console.WriteLine("Promoted >" + e.Name);

}

}

}

}

public class outworks

{

public List<Employee> outworksEmployees;

public outworks()

{

outworksEmployees = new List<Employee>()

{

new Employee { ID=1, Exp=10, Name="A", Salary=1000},

new Employee { ID=1, Exp=10, Name="b", Salary=2000},

new Employee { ID=1, Exp=10, Name="c", Salary=5000},

new Employee { ID=1, Exp=10, Name="d", Salary=4000},

};

}

public bool isPromotionApplication(Employee e)

{

return e.Salary < 4000 ? true : false;

}

}

public class WNS

{

public List<Employee> WNSEmployees;

public WNS()

{

WNSEmployees = new List<Employee>()

{

new Employee { ID=1, Exp=10, Name="WA", Salary=1000},

new Employee { ID=1, Exp=11, Name="Wb", Salary=2000},

new Employee { ID=1, Exp=12, Name="Wc", Salary=5000},

new Employee { ID=1, Exp=13, Name="Wd", Salary=4000},

};

}

public bool isPromotionApplication(Employee e)

{

return e.Exp >11 ? true : false;

}

}

static void Main(string[] args)

{

outworks ou = new outworks();

Employee employee = new Employee();

employee.Promoteemployee(ou.outworksEmployees, ou.isPromotionApplication);

WNS wNS = new WNS();

employee.Promoteemployee(wNS.WNSEmployees, wNS.isPromotionApplication);

}

}

}

**Event :**

**An event**  is notification send by object if an action performed. It follow observer design pattern . In .net button class has delegate & Click event is of that type , Page class which use those delegate have to defined OnClick event to raise those event.

E.g.

1. SSKPayament have a method which send mail after Payment completed initially it call sendMail method in last line of code

2. Then client want to send SMS as well after payment completed , so we need to change in Dopayment method which is wrong approach.

Inbelow code if new condition event comes in then we have to attach those event only , no need to change in dopayment method.

using System;

namespace DelegateEventExpl

{

public delegate void PayamentNotification(object sender, EventArgs e);

public class SSKPayament

{

public event PayamentNotification onPaymentCompleted;

public void doPayment()

{

Console.WriteLine("Do payament");

if(onPaymentCompleted!=null)

{

onPaymentCompleted(this, EventArgs.Empty);

}

}

}

public class SMSNotifier

{

public void sendSMS(object sender, EventArgs e)

{

Console.WriteLine("Send Sms");

}

}

public class EmailNotifier

{

public void sendEmail(object sender, EventArgs e)

{

Console.WriteLine("Send Email");

}

}

class Program

{

static void Main(string[] args)

{

SSKPayament sSKPayament = new SSKPayament();

EmailNotifier emailNotifier = new EmailNotifier();

SMSNotifier sMSNotifier = new SMSNotifier();

sSKPayament.onPaymentCompleted += emailNotifier.sendEmail;

sSKPayament.onPaymentCompleted += sMSNotifier.sendSMS;

Console.WriteLine("Hello World!");

sSKPayament.doPayment();

}

}

}

// Here we don’t need to add code in SSKpayament class if further we want to mode functionality , only we need to attach event of same signature.

1. What is Anonymous type?

Type without name .it contains readonly property.

var Student ={ Rollno =1 , Name =”chandan” }

Student is anonymous type Student.rollno is readonly.

In Linq

Var Student = Students.where (x => x.Id=1). Select (x => new {

Rollno=1,

Name =”Chandan”

});

Anonymous type is derived from System.Object.

Compiler create Autogenerated class name for Anonymous type and according to value generate Property type .

1. Why not use instead of Singleton pattern.
2. Both uses single instance of class but Static instance generated when class load while Singleton object created when required.
3. We can not pass object of Static class in any method while we can pass object of singleton object.
4. Suppose we have created a singleton class and it have multiple uses in application in future it is require that those class can have more than one instance then we need to make constructor as public while in case of static we cann’t do , we have re write code again

e.g.

1. . What is dependency Inversion principal ?

High level module can not depend on low level module . Both should be loosely coupled

Business Access layer should not initialize DAL layer both should depend on abstraction . (Abstarct /Interface)

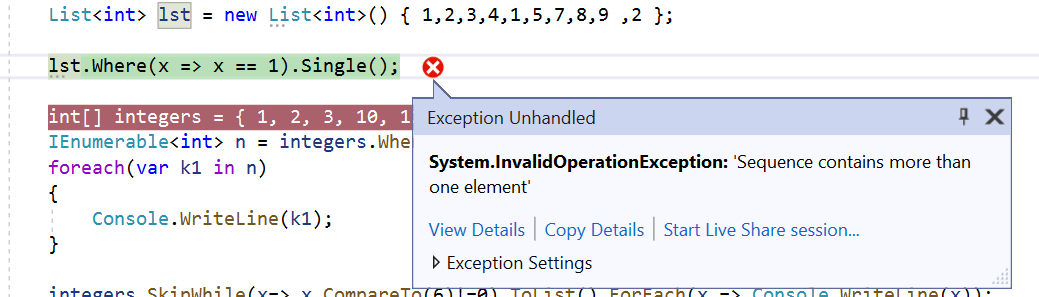
1. Types of DI ( Dependency Injection ) or how many ways we can use this.

[Constructor , Method , Property ]

1. Constractor : when instance required at multiple places
2. Method : when instance is required only that method
3. Prooperty : when instance required at multiple place but we don’t want our class have parametrized constructor
4. Difference between Single and First

Both will give single result

Single will return error if there are more than one element found, while first will return first element in sequnce



1. What is Factory pattern .

Abstracting the process of object creation . Factory class method will decide what type of object will be created .Instead client create that.

Creditcard = { Platinum , Silver , Gold }

ICICI want to initiate card delivery as per Employee salary .

Abstract class Creditcard

{

Public abstract void show();

}

Public class Platinum : Creditcard

{

Public override void show(){}

}

Public class Silver : Creditcard

{

Public override void show(){}

}

Public class Gold: CreditCard

{

Public override void show(){}

}

Public class FactoryCard

{

Public CreditCard getcardInstance(int type)

{

Switch(type)

{

Case 1: return new Paltinum();

Case 2: return new Silver();

Case 3: return new Gold();

}

}

}

// At client end

Factory f = new Factory();

f.getInstance(1).show();

\*\*\* Now suppose all sub class again have different set of bevhavior and their different set of mechanism then will create another layer and create abstract factory of different type .

1. What is MediaType formatter ?

Media type formatter are classed which serialize the request and response as per header .

At client At server

XML MediaTypeFormatter Connvert Request into CLR object

Content -type : application/xml

Accept: application/json

JSONMediaTypeformatter : Convert CLR Object to JSON Object

There are various types of Mediatype formatter : { JSONMediaTypeFormatter , XMLMediaTypeFormatter, FormUrlEncodedMediaTypeFormatter , JqueryMVCFormUrlEncodedFormatter , BSONMediaypeFormatter }

We can create our own custom MediaType Formatter

Public class PersonCSvFormatter : BufferMediaTypeFormatter

{

Public override bool CanReadType(Type type)

{

Return false;

}

Public override bool CanWriteType(Type type)

{

Return typeof(IEnumerable<Person>).IsAccessFrom(type);

}

}

1. What is VideModel and View

Model is set of information that represent Domain and ViewModel is information that we want to display on UI.

1. How Exception handle in MVC
2. Try catch finally
3. Overriding OnException method
4. [HandleError] attribute
5. Setting Global Exception handling filter
6. Handle on Application\_error
7. Extending HandleErrorAttribute

B .)

Public ActionResult Index()

{

Int a=1;

Int b=0;

Int c= a/b;

Return View();

}

Protected override void OnException(ExceptionContext filterContext)

{

Exception e = filterContext.Exception;

filterContext.HandledError =true;

filterContext.Result=new ViewResult()

{

Name =”Error” // CShtml should be defined in shared layout folder

}

}

C.)

[HandleError]

Public ActionResult Index()

{

Int a=1;

Int b=0;

Int c= a/b;

Return View();

}

<CustomError mode=”On”/>

Or

[HandleError]

[HandleError(ExceptionType=typeof(DivideByZeror), View=”Error1”)]

[HandleError(ExceptionType=typeof(ArgumentOutofRange), View=”Error2”)]

Public ActionResult Index()

{

Int a=1;

Int b=0;

Int c= a/b;

Return View();

}

d.)

Setting Global Filter attributes

GlobalFilters.Filters.Add(new HandleErrorAttribute() { view =”GlobalErro” }); on Application\_error Event

e.)

Cutom Filter

Public class CutomException : IFilterAttribute , IExceptionFilter

{

Public override void OnException(ExceptionContext filterContext)

{

Exception e = filterContext.Exception;

filterContext.HandledError =true;

filterContext.Result=new ViewResult()

{

Name =”Error” // CShtml should be defined in shared layout folder

}

}

}

& then

[CutomException]

Public ActionResult Index()

{

Int a=1;

Int b=0;

Int c= a/b;

Return View();

}

1. How to Improve DB performance in SQL server
2. What difference between Rank and Dense\_Rank
3. What Is CTE
4. What is Trigger