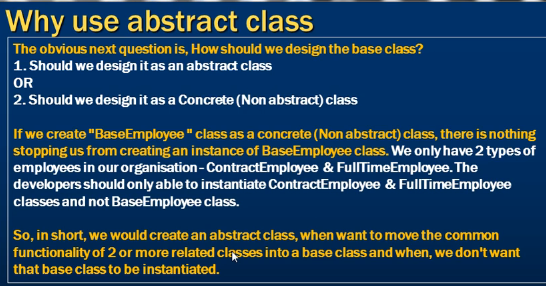
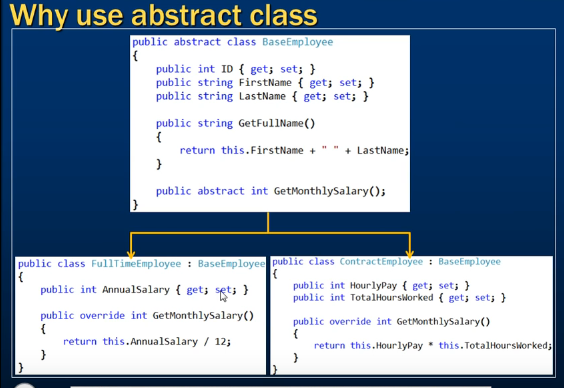
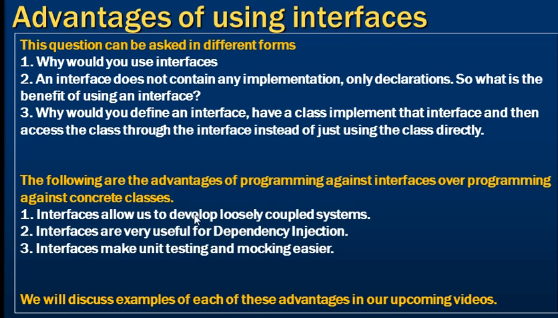
# Abstract class

Common code are duplicated in two classes ContractEmployee and FullTimeEmployee





# Interfaces



# Abstract classes vs Interfaces

Короткое различие.

Абстрактный класс — это класс, у которого не реализован один или больше методов (некоторые языки требуют такие методы помечать специальными ключевыми словами).

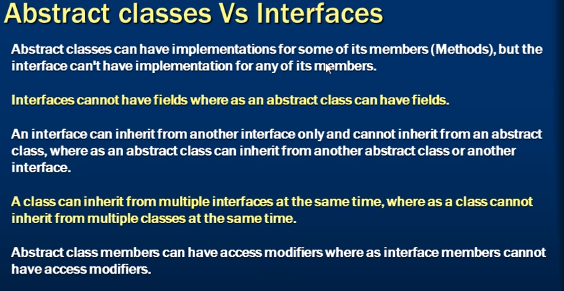
Интерфейс — это абстрактный класс, у которого **ни один** метод не реализован, все они публичные и нет переменных класса.

Интерфейс нужен обычно когда описывается только интерфейс (тавтология). Например, один класс хочет дать другому возможность доступа к некоторым своим методам, но не хочет себя «раскрывать». Поэтому он просто реализует интерфейс.

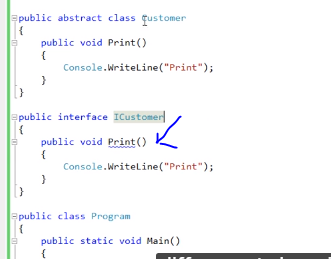
Абстрактный класс нужен, когда нужно семейство классов, у которых есть много общего. Конечно, можно применить и интерфейс, но тогда нужно будет писать много идентичного кода.

В некоторых языках (С++) специального ключевого слова для обозначения интерфейсов нет.

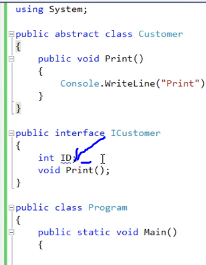
Можно считать, что любой интерфейс — это уже абстрактный класс, но не наоборот.



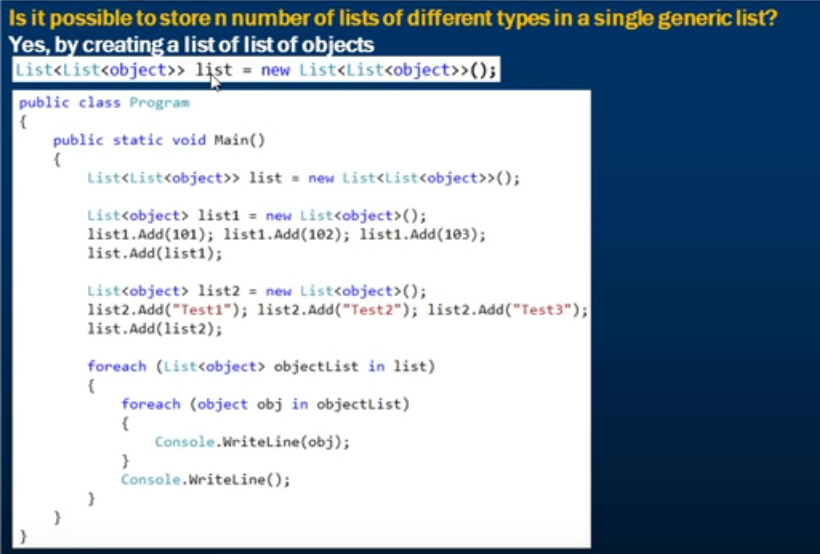
Interfaces cannot have implementation



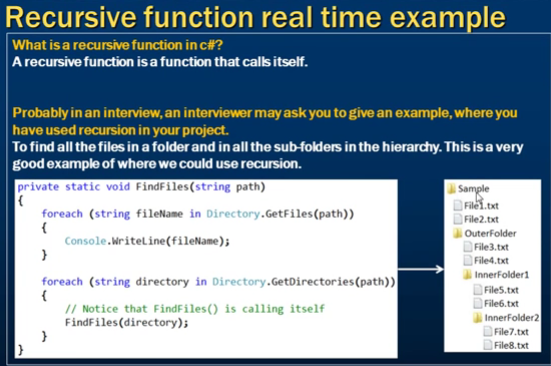
Interfaces cannot have field



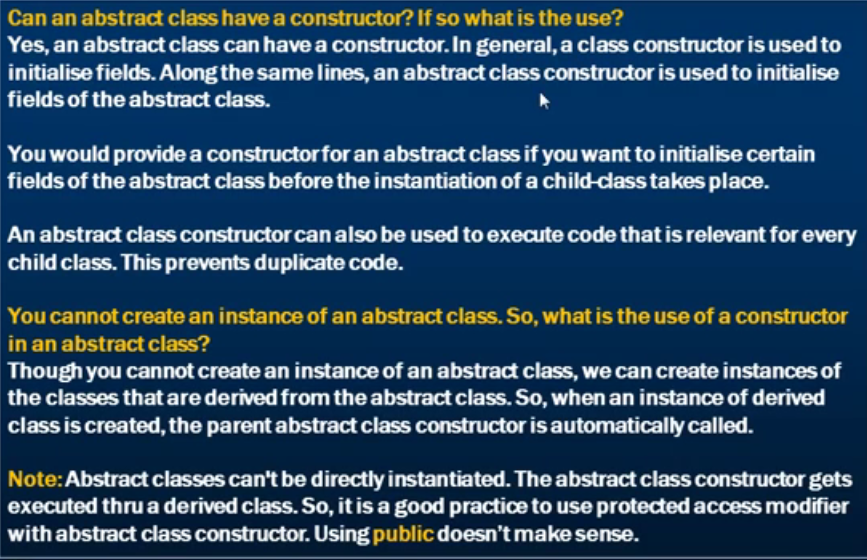
# Storing different types lists in list

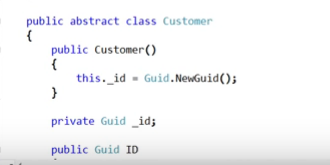


# Recursive

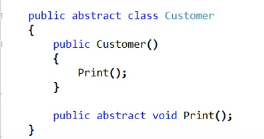


# Abstract class constructor

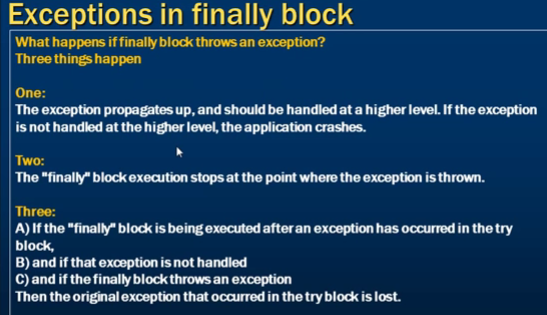




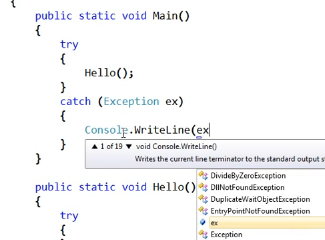
Calling an abstract method from an abstract class



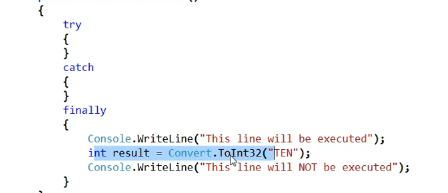
# Finally block exception



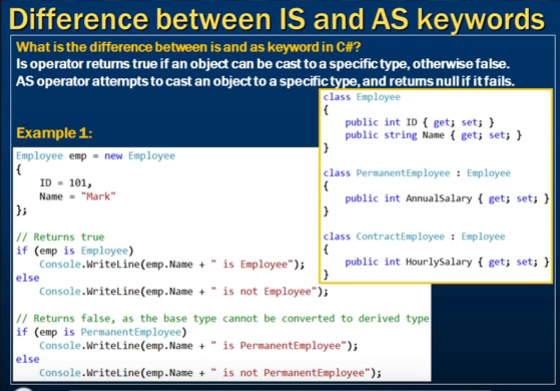
Higher level catching

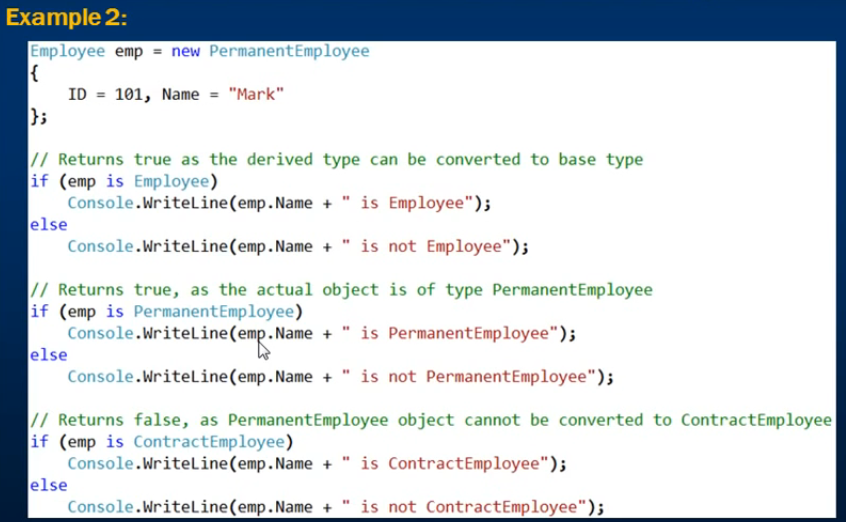


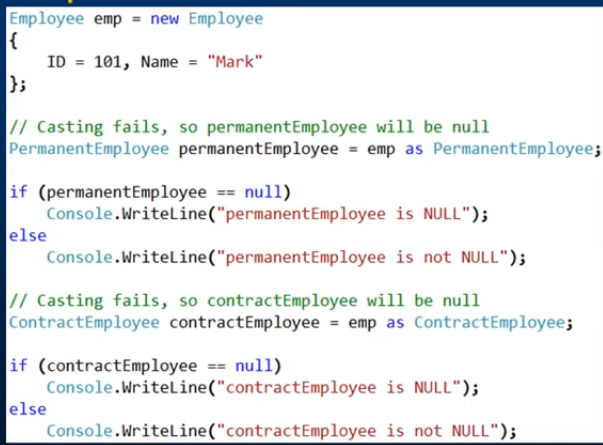
Stop at the point where the exception throw

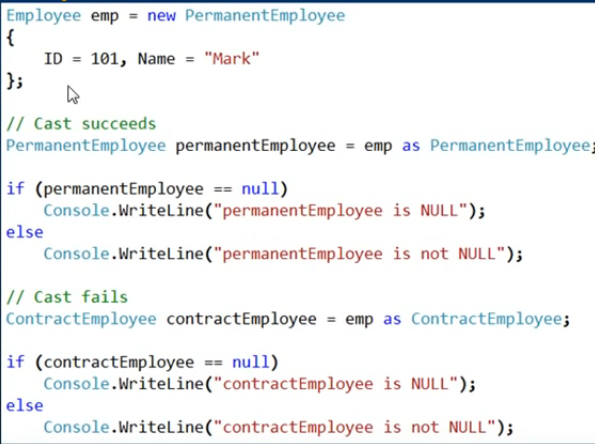


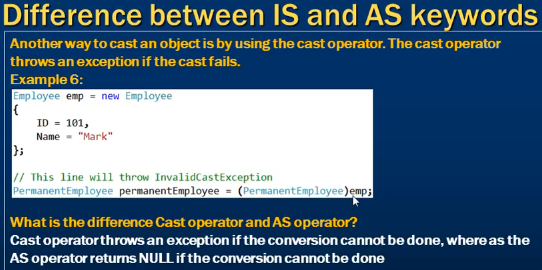
# Is and As differences



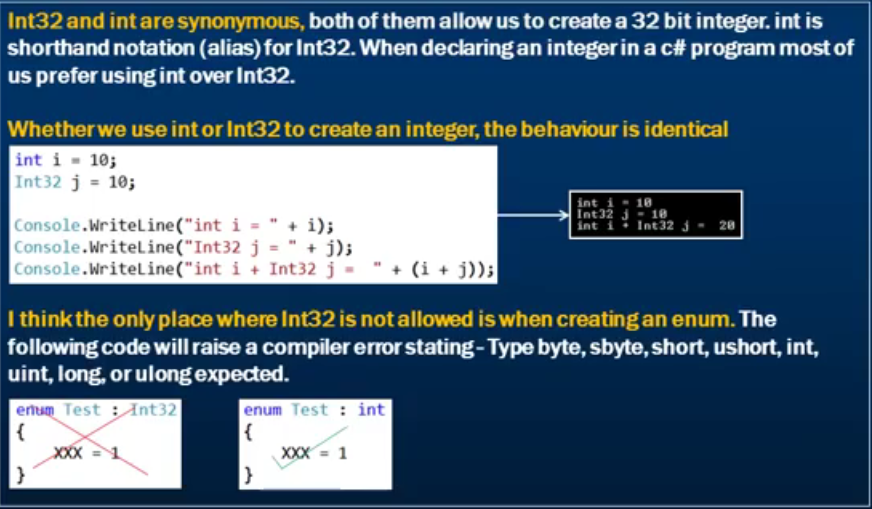




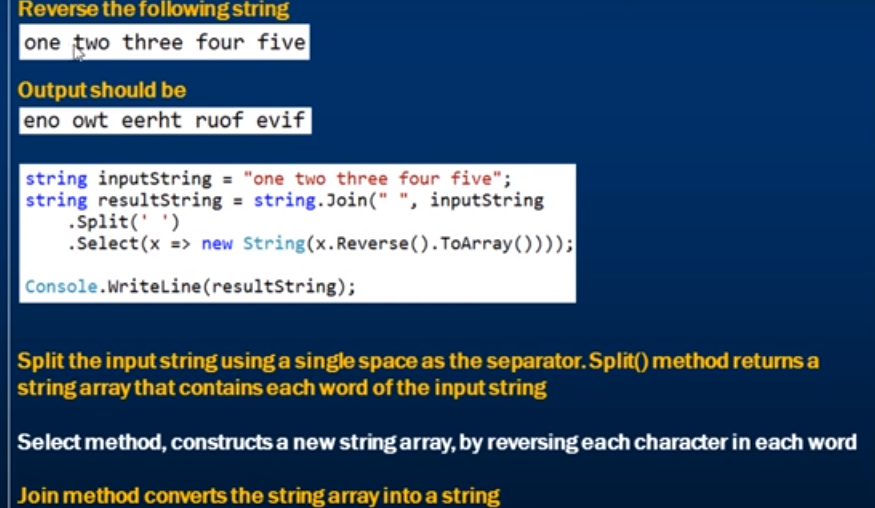




# Differences between Int32 and int



# Reverse each word in string



# Test c# beginer

1. Which of the followings is not allowed in C# as access modifier?

public

friend

internal

protected

 Correct!

2. In the C# code below, what is this[int i]?   
  
class MyClass   
{   
// ...   
  
public string this[int i]   
{   
get{ return arr[i];}   
set{ arr[i] = value; }   
}   
}

Property

Event

Indexer

Delegate

 Correct!

3. Which of the following C# keywords has nothing to do with multithreading?

async

await

sealed

lock

 Correct!

4. Find an invalid expression among the following C# Generics examples.

class A where T : class, new()

class A where T : struct, IComparable

class A where T : class, struct

class A where T : Stream where U : IDisposable

 Correct answer is 3

5. new keyword in C# is used to creat new object from the type. Which of the followings is not allowed to use new keyword?

Class: var a = new Class1();

Interface : var a = new IComparable();

Struct : var a = new Struct1();

C# object : var a = new object();

 Correct!

6. In the example below, button1 is an object of Button class in WinForms. Which one is a wrong expression as a click event handler?

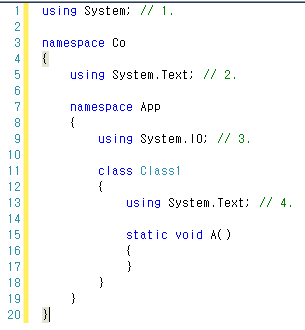
button1.Click += new System.EventHandler(button1\_Click);

button1.Click += delegate { MessageBox.Show("Click"); };

button1.Click += delegate(EventArgs e){MessageBox.Show("Click");};

button1.Click += (s, e) => MessageBox.Show("Click");

 Correct answer is 3

7. In the C# example below, which using statement is wrong?   


1

2

3

4

 Correct!

8. What is the output of this C# code?   
  
int? i = 8 >> 5;   
int? j = i > 0 ? i : null;   
var a = j ?? int.MinValue;   
Console.WriteLine(a);

1

null

0

-2147483648

 Correct!

9. Find a correct statement about C# exception

C# exception occrs at compile time

C# exception occrs at linking time

C# exception occrs at JIT compile time

C# exception occrs at run time

 Correct!

10. Find an invalid Main() method prototype, which is entry point in C#?

public static void Main()

public static int Main()

public static void Main(string[] s)

public static long Main(string[] args)

 Correct answer is 4

# Advance c# test

1. The following C# code is using C# Generics. Which is an incorrect explanation?   
  
T t = default(T);

If T is int type, variable t has 0

If T is a reference type, variable t has null

If T is string type, variable t has an empty string

If T is bool type, variable t has false

 Correct!

2. C# / .NET supports various built-in data structures. Which of the following data structures does not exist as built-in?

Array

D-Array

Binary Tree

Stack

Linked List

 Correct answer is 3

3. Which of the following interfaces should be implemented to use LINQ to Objects?

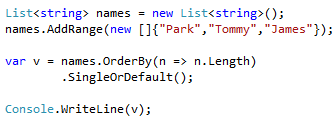
IEnumerable

IEnumerable

ICollection

ICollection

 Correct answer is 2

4. What is the result of the following C# code?   


null will be displayed in console

4 will be displayed in console

Will cause compile error

Runtime exception will occur

 Correct!

5. Find an invalid example of using C# var

var a = 3.141592;

var a = null;

var a = db.Stores;

var a = db.Stores.Single(p => p.Id == 1);

 Correct answer is 2

he following restrictions apply to implicitly-typed variable declarations:

* var can only be used when a local variable is declared and initialized in the same statement; the variable cannot be initialized to null, or to a method group or an anonymous function.
* var cannot be used on fields at class scope.
* Variables declared by using var cannot be used in the initialization expression. In other words, this expression is legal: int i = (i = 20); but this expression produces a compile-time error: var i = (i = 20);
* Multiple implicitly-typed variables cannot be initialized in the same statement.
* If a type named var is in scope, then the var keyword will resolve to that type name and will not be treated as part of an implicitly typed local variable declaration.

6. In the following C# example, variable a is string. Find one that either is wrong or returns a different result.

a = a ?? "";

a = a == null ? "" : a;

a = (a is null) ? "" : a;

if (a == null) a = "";

 Correct answer is 3

7. When you want to provide other resources for other cultures or languages, which assembly should you create?

Public Assembly

Private Assembly

Shared Assembly

Satellite Assembly

 Correct answer is 4

A definition from MSDN says something like this: "A .NET Framework assembly containing resources specific to a given language. Using satellite assemblies, you can place the resources for different languages in different assemblies, and the correct assembly is loaded into memory only if the user elects to view the application in that language

8. Which of the followings does not allow you to use C# static keyword?

(Method) static void Run() {}

(Property) static int Prop {get; set;}

(Field) static int \_field;

(Class) static class MyClass {}

(Constructor) static MyClass() {}

(Destructor) static ~MyClass() {}

(Event) static event EventHandler evt;

 Correct!

9. In C#, what is similar to C++ function pointer?

Event

Interface

Delegate

Method

 Correct!

10. Which of the following C# methods is not valid? (method body elided)

public void Set(dynamic o) {}

public dynamic Get() {}

private var GetData() {}

protected override int[] A() {}

 Correct!

Professional c# test

1. Which of the following statements is not valid to create new object in C#?

var a = new Int32();

var a = new String();

var a = new IComparable();

var a = new [] {0};

 Correct!

2. If you run C# executable file multiple times, multiple processes are created. If you want to have only one application process even if you launch multiple times, what can you use?

Semaphore

Mutex

Critical Section

C# lock

 Correct!

3. Which of the following operators cannot use operator overloading?

operator ++

operator &

operator ||

operator true

 Correct answer is 3

|  |  |
| --- | --- |
| [&&](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/conditional-and-operator), [||](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/conditional-or-operator) | The conditional logical operators cannot be overloaded, but they are evaluated using & and |, which can be overloaded. |

4. In multithread programming, which of the followings is not using Thread Pool?

BackgroundWorker class

Asynchronous delegate

Thread class

Task class

 Correct answer is 3

5. Class A has [Serializable()] attribute. When is [Serializable] checked?   
  
[Serializable()]   
class A { }

At C# compile time

At CLR runtime

At JIT compile time

At Linking

 Correct answer is 2

6. The followings are some examples of integer arrays. Which expression is not valid in C#?

int[] a = new int[10];

int[][] c = new int[10][];

int[][][] cc = new int[10][2][];

int[,] b = new int[10, 2];

int[, , ,] d = new int[10, 2, 2, 2];

 Correct!

7. Which of the following statements is true about C# anonymous type?

Anonymous type can add new property once it is created

Anonymous type can add an event

You can use a delegate for a method in anonymous type

Anonymous type is an immutable type

 Correct answer is 4

8. What is the result of variable a and b?   
  
var a = 5L == 5.0F;   
var b = 24L / 5 == 24 / 5d;

a=true, b=true

a=true, b=false

a=false, b=true

a=false, b=false

 Correct answer is 2

9. When defining a class using C# Generics, which of the followings is invalid?

class MyClass where T : struct

class MyClass where T : class

class MyClass where T : IComparable

class MyClass where T : MyBase

All of the above are correct

 Correct!

10. Which of the following statements is incorrect about C# delegate?

C# delegate supports multicast

C# delegate is considered as a technical basis of C# event

C# delegate can be used when passing a reference to a method

C# delegate can not use +=, -= operators

 Correct answer is 4

# C# interview

**4) What is the difference between “dispose” and “finalize” variables in C#?**

* Dispose - This method uses interface – “IDisposable” interface and it will free up both managed and unmanaged codes like – database connection, files etc.
* Finalize - This method is called internally unlike Dispose method which is called explicitly. It is called by garbage collector and can’t be called from the code.
* **16) Can we execute multiple catch blocks in C#?**
* No. Once any exception is occurred it executes specific exception catch block and the control comes out.
* **20) Can we have only “try” block without “catch” block in C#?**
* Yes we can have only try block without catch block but we have to have finally block.

**21) List out two different types of errors in C#?**

Below are the types of errors in C# -

* Compile Time Error
* Run Time Error

**23) Mention the assembly name where System namespace lies in C#?**

Assembly Name – mscorlib.dll

**24) What are the differences between static, public and void in C#?**

* Static classes/methods/variables are accessible throughout the application without creating instance. Compiler will store the method address as an entry point.
* Public methods or variables are accessible throughout the application.
* Void is used for the methods to indicate it will not return any value.
* **27) Can we use “this” inside a static method in C#?**
* No. We can’t use “this” in static method.

**30) Can we override private virtual method in C#?**

No. We can’t override private virtual methods as it is not accessible outside the class.

**31) Explain access modifier – “protected internal” in C#?**

“protected internal” can be accessed in the same assembly and the child classes can also access these methods.

**32) In try block if we add return statement whether finally block is executed in C#?**

Yes. Finally block will still be executed in presence of return statement in try block.

**34) Explain String Builder class in C#?**

This will represent the mutable string of characters and this class cannot be inherited. It allows us to Insert, Remove, Append and Replace the characters. “ToString()” method can be used for the final string obtained from StringBuilder. For example,

StringBuilder TestBuilder = new StringBuilder("Hello");  
TestBuilder.Remove(2, 3); // result - "He"  
TestBuilder.Insert(2, "lp"); // result - "Help"  
TestBuilder.Replace('l', 'a'); // result - "Heap"

**36) What is the difference between methods – “System.Array.Clone()” and “System.Array.CopyTo()” in C#?**

* “CopyTo()” method can be used to copy the elements of one array to other.
* “Clone()” method is used to create a new array to contain all the elements which are in the original array.

**39) List out some of the exceptions in C#?**

Below are some of the exceptions in C# -

* NullReferenceException
* ArgumentNullException
* DivideByZeroException
* IndexOutOfRangeException
* InvalidOperationException
* StackOverflowException etc.

**40) Explain Generics in C#?**

Generics in c# is used to make the code reusable and which intern decreases the code redundancy and increases the performance and type safety.   
Namespace – “System.Collections.Generic” is available in C# and this should be used over “System.Collections” types.

**43) What are the types of delegates in C#?**

Below are the uses of delegates in C# -

* Single Delegate
* Multicast Delegate
* Generic Delegate

**44) What are the three types of Generic delegates in C#?**

Below are the three types of generic delegates in C# -

* Func
* Action
* Predicate

**45) What are the differences between events and delegates in C#?**

Main difference between event and delegate is event will provide one more of encapsulation over delegates. So when you are using events destination will listen to it but delegates are naked, which works in subscriber/destination model.

Объявление события добавляет слой абстракции и защиты экземпляра делегата. Эта защита не позволяет клиентам делегата сбросить делегат и его список вызовов и позволяет добавлять или удалять цели из списка вызовов.

**46) Can we use delegates for asynchronous method calls in C#?**

Yes. We can use delegates for asynchronous method calls.

**5. What is the difference between a struct and a class in C#?**  
  
**Answer:**

Class and struct both are the user defined data type but have some major difference:  
 **Struct**

* The struct is value type in C# and it inherits from System.Value Type.
* Struct is usually used for smaller amounts of data.
* Struct can’t be inherited to other type.
* A structure can't be abstract.
* No need to create object by new keyword.
* Do not have permission to create any default constructor.

**Class**

* The class is reference type in C# and it inherits from the System.Object Type.
* Classes are usually used for large amounts of data.
* Classes can be inherited to other class.
* A class can be abstract type.
* We can’t use an object of a class with using new keyword.
* We can create a default constructor.

For more details just go with the following link:

**47) What are the uses of delegates in C#?**

Below are the list of uses of delegates in C# -

* Callback Mechanism
* Asynchronous Processing
* Abstract and Encapsulate method
* Multicasting

**48) What is Nullable Types in C#?**

Variable types does not hold null values so to hold the null values we have to use nullable types. So nullable types can have values either null or other values as well.

Eg: Int? mynullablevar = null;

**50) What is the difference between “as” and “is” operators in C#?**

* “as” operator is used for casting object to type or class.
* “is” operator is used for checking the object with type and this will return a Boolean value.
* 51) Define Multicast Delegate in C#?
* A delegate with multiple handlers are called as multicast delegate. The example to demonstrate the same is given below
* public delegate void CalculateMyNumbers(int x, int y);  
  int x = 6;  
  int y = 7;  
  CalculateMyNumbers addMyNumbers = new CalculateMyNumbers(FuncForAddingNumbers);  
  CalculateMyNumbers multiplyMyNumbers = new CalculateMyNumbers(FuncForMultiplyingNumbers);  
  CalculateMyNumbers multiCast = (CalculateMyNumbers)Delegate.Combine (addMyNumbers, multiplyMyNumbers);  
  multiCast.Invoke(a,b);

**53) Is C# code is unmanaged or managed code?**

C# code is managed code because the compiler – CLR will compile the code to Intermediate Language

**55) Explain Hashtable in C#?**

It is used to store the key/value pairs based on hash code of the key. Key will be used to access the element in the collection. For example,

Hashtable myHashtbl = new Hashtable();  
myHashtbl.Add("1", "TestValue1");  
myHashtbl.Add("2", "TestValue2");

**60) Write a sample code to write the contents to text file in C#?**

Below is the sample code to write the contents to text file –

Using System.IO;  
File.WriteAllText(”mytextfilePath”, “MyTestContent”);

**62) Explain Partial Class in C#?**

Partial classes concept added in .Net Framework 2.0 and it allows us to split the business logic in multiple files with the same class name along with “partial” keyword.

**63) Explain Anonymous type in C#?**

This is being added in C# 3.0 version. This feature enables us to create an object at compile time. Below is the sample code for the same –

Var myTestCategory = new { CategoryId = 1, CategoryName = “Category1”};

**64) Name the compiler of C#?**

C# Compiler is – CSC.

**66) Explain Copy constructor in C#?**

If the constructor contains the same class in the constructor parameter then it is called as copy constructor.

class MyClass  
{  
 public string prop1, prop2;  
 public MyClass(string a, string b)  
 {  
 prop1 = a;  
 prop2 = b;  
 }  
   
 public MyClass(MyClass myobj) // Copy Constructor  
 {  
 prop1 = myobj.prop1;  
 prop2 = myobj.prop2;  
 }  
}

67) Explain Static constructor in C#?

If the constructor is declared as static then it will be invoked only once for all number of instances of a class. Static constructor will initialize the static fields of a class.

class MyClass  
{  
 public string prop1, prop2;  
 public MyClass(string a, string b)  
 {  
 prop1 = a;  
 prop2 = b;  
 }

Static MyClass()  
 {  
 Console.WriteLine(“Static Constr Test”);  
 }  
 public MyClass(MyClass myobj) // Copy Constructor  
 {  
 prop1 = myobj.prop1;  
 prop2 = myobj.prop2;  
 }  
}

**71) Explain Attributes in C#?**

* Attributes are used to convey the info for runtime about the behavior of elements like – “methods”, “classes”, “enums” etc.
* Attributes can be used to add metadata like – comments, classes, compiler instruction etc.

**72) List out the pre defined attributes in C#?**

Below are the predefined attributes in C# -

* Conditional
* Obsolete
* Attribute Usage

**73) What is Thread in C#?**

Thread is an execution path of a program. Thread is used to define the different or unique flow of control. If our application involves some time consuming processes then it’s better to use Multithreading., which involves multiple threads.

**74) List out the states of a thread in C#?**

Below are the states of thread –

* Unstarted State
* Ready State
* Not Runnable State
* Dead State

**75) Explain the methods and properties of Thread class in C#?**

Below are the methods and properties of thread class –

* CurrentCulture
* CurrentThread
* CurrentContext
* IsAlive
* IsThreadPoolThread
* IsBackground
* Priority

В C# применяются следующие модификаторы доступа:

* **public**: публичный, общедоступный класс или член класса. Такой член класса доступен из любого места в коде, а также из других программ и сборок.
* **private**: закрытый класс или член класса. Представляет полную противоположность модификатору public. Такой закрытый класс или член класса доступен только из кода в том же классе или контексте.
* **protected**: такой член класса доступен из любого места в текущем классе или в производных классах. При этом производные классы могут располагаться в других сборках.
* **internal**: класс и члены класса с подобным модификатором доступны из любого места кода в той же сборке, однако он недоступен для других программ и сборок (как в случае с модификатором public).
* **protected internal**: совмещает функционал двух модификаторов. Классы и члены класса с таким модификатором доступны из текущей сборки и из производных классов.
* **private protected**: такой член класса доступен из любого места в текущем классе или в производных классах, которые определены в той же сборке.

**89) What is Polymorphism in C# ?**

The ability of a programming language to process objects in different ways depending on their data type or class is known as Polymorphism. There are two types of polymorphism

* Compile time polymorphism. Best example is Overloading
* Runtime polymorphism. Best example is Overriding