C# Interview Questions And Answers Pdf

**1. What are the fundamental principles of OO programming?**  
**Answer:**As a developer, you might be tempted to answer that it comprises things like Encapsulation, Polymorphism, Abstraction, and Inheritance. Although this is true, it doesn’t really explain the fundamental core of what OO is and what its benefits are.

Principles are crucial but they are not the most important aspect of what OO actually is. What is really important is to understand in what grounds OO is built upon, or in other words, what are the foundations of OO programming.

The two most fundamental core concepts on which OO has been built upon in C# are this pointer and Dynamic Dispatch.

Obviously, there are principles like Encapsulation, Polymorphism, Abstraction, and Inheritance, but these are the consequence and not the generating force behind the OO paradigm in C#.

**2. Explain Partial Class in C#?**  
**Answer:**Partial classes concept added in .Net Framework 2.0 and it allows us to split the business logic into multiple files with the same class name along with “partial” keyword.

**3. What is Thread in C#?**  
**Answer:**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.

**4. What does it mean?**  
**Answer:**Object Pool is nothing but a container of objects that are ready for use. Whenever there is a request for a new object, the pool manager will take the request and it will be served by allocating an object from the pool.

**5. How does it work?**  
**Answer:**We are going to use a Factory pattern for this purpose. We will have a factory method, which will take care of the [creation](https://en.wikipedia.org/wiki/.net) of objects. Whenever there is a request for a new object, the factory method will look into the object pool (we use Queue object). If there is any object available within the allowed limit, it will return the object (value object), otherwise, a new object will be created and give you back.

**6. What is the difference between early binding and late binding in C#?**  
**Answer:**Early binding and late binding are the concepts of polymorphism. There are two types of polymorphism in C#.

Compile Time Polymorphism: It is also known as early binding.  
Run Time Polymorphism: It is also known as late binding or method overriding or dynamic polymorphism.

**7. What is ArrayList?**  
**Answer:**ArrayList is a dynamic array. You can add and remove the elements from an ArrayList at runtime. In the ArrayList, elements are not automatically sorted.

**8. What do you mean by the delegate in C#?**  
**Answer:**Delegates are type-safe pointers, unlike function pointers as in C++. Delegate is used to represent the reference of the methods of some return type and parameters.

**9. What is a constructor?**  
**Answer:**A constructor is a class member executed when an instance of the class is created. The constructor has the same name as the class, and it can be overloaded via different signatures. Constructors are used for initialization chores.

**10. Why are strings in C# immutable?**  
**Answer:**Immutable means string values cannot be changed once they have been created. Any modification to a string value results in a completely new string instance, thus an inefficient use of memory and extraneous garbage collection. The mutable System.Text.StringBuilder class should be used when string values will change.

**11. What is object pool in .Net?**  
**Answer:**An object pool is a container of ready to use objects. It reduces the overhead of creating a new object.

**12. Questions on Looping Statements**  
**The section contains questions on if, while, do-while, for, switch, continue and goto looping statements?**  
**Answer:**

* IF Statements
* Switch Statements
* For Loop Statements While Loop Statements
* Do While Loop Statements
* Continue, Goto statements

**13. What is Garbage Collection?**  
**Answer:**Garbage Collection is a process of releasing memory automatically occupied by objects which are no longer accessible.  
Describe the basic construction of a C# program. Write a simple program that outputs “Hello World” to the console.

A typical C# program consists of a namespace declaration, a class, methods, attributes, the main method, statements, expressions, and comments. A potential example for printing “Hello World” to the console is detailed below.

using System;  
namespace HelloWorldApplication  
class HelloWorld  
static void Main(string[] args)

Console.WriteLine(“Hello World”);  
Console.ReadKey();

**14. Questions and Answers on Object-Oriented Concepts?**  
**Answer:**The section contains questions and answers on polymorphism, structures, enumeration, inheritance, method overloading and method overriding, abstract class and interfaces implementation.

* Polymorphism
* Structures
* Enumerations
* Fundamentals of Inheritance
* Inheritance Implementation
* Method Overloading
* Method Overriding Constructor Overloading
* Abstract Class & Methods
* Interfaces Introduction
* Interfaces Implementation
* Introduction of Overloaded Operators
* Recursion

**15. What is the difference between disposing of () and finalize() methods in C#?**  
**Answer:**The dispose of the () method is explicitly called by user to free unmanaged resources such as files, database connections, etc whereas finalizing () method is implicitly called by the garbage collector to free unmanaged resources like files, database connections, etc.

The dispose of () method belongs to the IDisposable interface whereas finalize() method belongs to the Object class.

**16. What is a delegate in C?**  
**Answer:**A delegate in C# is an object that holds the reference to a method. It is like a function pointer in C++.

**17. What is the difference between dynamic type variables and object type variables in C#?**  
**Answer:**The difference between dynamic and object type variables has to do with when the type checking takes place during the application lifecycle. Dynamic type variables handle type checking at run time, while object type variables handle type checking during compile time.

**18. Explain nullable types in C#?**  
**Answer:**Nullable types are data types that, in addition to their normal values, also contain a defined data type for null. Nullable types exist to help integrate C#, which generally works with value types, and databases, which often use null values. You can declare a nullable type in C# using the following syntax:

**19. What is Managed and Unmanaged code?**  
**Answer:**Managed code is a code which is executed by CLR (Common Language Runtime) i.e all application code based on .Net Platform. It is considered as managed because of the .Net framework which internally uses the garbage collector to clear up the unused memory.

Unmanaged code is any code that is executed by application runtime of any other framework apart from .Net. The application runtime will take care of memory, security and other performance operations.

**20. Questions and Answers on Data Types, Variables and Operators?**  
**Answer:**The section contains questions and answers on integer, float, decimal and character data types, variable initialization, their scope and lifetime, arithmetic, relational, logical, bitwise and conditional operators.

* Integer Data Types
* Floating and Decimal Data Types
* Char Types and String Literals
* Initialization of Variables
* Scope and Lifetime of Variables Type Conversion in
* Expressions
* Arithmetic Operators
* Relational and Logical Operators
* Bit-wise and Conditional Operators

**21. Questions and Answers on Miscellaneous topics?**  
**Answer:**The section contains questions and answers on the basics of pointers and their operation, accessor controls, and string formatting.

* Unsafe Code & Pointers Basics
* Pointers Operation – 1
* Pointers Operation – 2
* Accessor Controls of Class Introduction of String Formatting
* String Formatting – 1
* String Formatting – 2
* If you like to learn C# thoroughly, you should attempt to work
* on the complete set of C# questions and answers mentioned above. It will immensely help anyone trying to crack a C#
* code or an interview.
* Here’s the list of Best Reference Books in C# Programming Language.

**22. Questions and Answers on Reflections, Multithreaded Programming, Collection Classes, and Mathematical Functions?**  
**Answer:**The section contains questions and answers on collection framework, mathematical classes, rounding functions, iterators, and multithreaded programming.

* Introduction of Reflections
* Collection Classes
* Maths Class
* Rounding Functions in C# Multi-threaded Programming – 1
* Multi-threaded Programming – 2
* Iterators

**23. what is the solution if you need to manipulate sets of items?**  
**Answer:**One solution would be to create a variable for each item in the set but again this leads to a different problem.  ( [devops training online](https://svrtechnologies.com/devops-training/" \t "_blank) )

**24. How many variables do you need?**  
**Answer:**So in this situation Arrays provide mechanisms that solve the problem posed by these questions. An array is a collection of related items, either value or reference type. In C# arrays are immutable such that the number of dimensions and size of the array are fixed.

**Arrays Overview**

An array contains zero or more items called elements. An array is an unordered sequence of elements. All the elements in an array are of the same type (unlike fields in a class that can be of different types). The elements of an array accessed using an integer index that always starts from zero. C# supports single-dimensional (vectors), multidimensional and jagged arrays.

Elements are identified by indexes relative to the beginning of the arrays. An index is also commonly called indices or subscripts and is placed inside the indexing operator ([]). Access to array elements is by their index value that ranges from 0 to (length-1).

**Array Properties**

The length cannot be changed once created.  
Elements are initialized to default values.  
Arrays are reference types and are instances of System. Array.  
Their number of dimensions or ranks can be determined by the Rank property.  
An array length can be determined by the GetLength() method or Length property.

**25. What is Parsing? How to Parse a Date Time String?**  
**Answer:**Parsing is converting a string into another data type.

**For Example:**

string text = “500”;

int num = int.Parse(text);

500 is an integer. So, the Parse method converts the string 500 into its own base type, i.e int.

Follow the same method to convert a DateTime string.  
string DateTime = “Jan 1, 2018”;  
DateTime parsed value = DateTime.Parse(DateTime).

**26. What are the collection types can be used in C#?**  
**Answer:**

* Below are the collection types in C# –
* ArrayList
* Stack
* Queue
* SortedList
* HashTable
* Bit Array

**27. Explain Attributes in C#?**  
**Answer:**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.

**28. Name some properties of Thread Class?**  
**Answer:**

**Few Properties of thread class are:**

IsAlive – contains value True when a thread is Active.  
Name – Can return the name of the thread. Also, can set a name for the thread.  
Priority – returns the prioritized value of the task set by the operating system.  
background – gets or sets a value which indicates whether a thread should be a background process or foreground.  
ThreadState– describes the thread state.

**29. List out the differences between Array and ArrayList in C#?**  
**Answer:**Array stores the values or elements of the same data type but ArrayList stores values of different data types.  
Arrays will use the fixed length but ArrayList does not use fixed-length like an array.

**30. Why use “using” in C#?**  
**Answer:**“Using” statement calls – “dispose of” method internally, whenever any exception occurred in any method call and in “Using” statement objects are read-only and cannot be reassignable or modifiable.

**31. Explain namespaces in C#?**  
**Answer:**Namespaces are containers for the classes. We will use namespaces for grouping the related classes in C#. “Using” keyword can be used for using the namespace in another namespace.

**32. List out two different types of errors in C#?**  
**Answer:**

* Below are the types of errors in C# –
* Compile Time Error
* Run Time Error

**33. What are the differences between static, public and void in C#?**  
**Answer:**Static classes/methods/variables are accessible throughout the application without creating an instance. The compiler will store the method address as an entry point.  
Public methods or variables are accessible throughout the application.  
The void is used for the methods to indicate it will not return any value.

**34. Do we get an error while executing the “finally” block in C#?**  
**Answer:**Yes. We may get an error in finally block.

**35. What is the difference between “out” and “ref” parameters in C#?**  
**Answer:**“out” parameter can be passed to a method and it need not be initialized whereas “ref” parameter has to be initialized before it is used.

**36. Explain Jagged Arrays in C#?**  
**Answer:**If the elements of an array is an array then it’s called a jagged array. The elements can be of different sizes and dimensions.

**37. What are the differences between events and delegates in C#?**  
**Answer:**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 the subscriber/destination model.

**38. What is a Generic Class?**  
**Answer:**Generics or Generic class is used to create classes or objects which do not have any specific data type. The data type can be assigned during runtime, i.e when it is used in the program.

**For Example:**

Generic Class

So, from the above code, we see 2 compare methods initially, to compare string and int.

In case of other data type parameter comparisons, instead of creating many overloaded methods, we can create a generic class and pass a substitute data type, i.e T. So, T acts as a datatype until it is used specifically in the Main() method.