|  |  |  |
| --- | --- | --- |
| ****C# Basics and Fundamentals**** | | |
| 1. **No.** | **Topics** | **Page** |
| 1 | What is C#? | 4 |
| 2 | Explain the main features of C# | 4 |
| 3 | What is the difference between value types and reference types in C#? | 4 |
| 4 | What is a nullable type in C#? | 4 |
| 5 | Explain the concept of boxing and unboxing. | 5 |
| 6 | What is the purpose of the var keyword in C# | 5 |
| 7 | What is the difference between const and readonly in C# | 5 |
| 8 | What are properties in C# | 6 |
| 9 | Explain the concept of indexers in C# | 6 |
| 10 | What is the purpose of the using statement in C# | 7 |
| 11 | Explain the concept of extension methods | 7 |
| 12 | Explain the static constructor |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**What is C#?**

* C# is a modern, high level programming langugue developed by Microsoft.
* It’s a part of .Net ecosystem, which is free, open source, cross platform framework that supports wide range of application development

**Explain the main features of C#**

* Object Oriented Programming - (Encapsulation, Inheritance, Polymorphism and Abstraction)
* Type Safety - enforces type constrains, helps us to perform operation on the compatible type reducing the runtime error
* Garbage Collection - automatic garbage collection, automatically remove the unused object from the memory. Reduces the developer’s effort of memory management
* Rich Library Support
* Cross Platform Development

**What is the difference between value types and reference types in C#?**

* Value Types are stored in the stack memory whereas ref type are stored in head memory only it’s address is stored in stack memory
* Stack is used in pace of small and immutable data and where performance is priority
* Heap is used in place of handling complex object, to share data across multiple places and places where null need to be handled

|  |  |
| --- | --- |
| **Stack** | **Heap** |
| Generally Faster | Heap allocation takes more time |
| Cannot be null unless made nullable | Can be null |
| Copies the Value | Copies the reference |

**What is nullable type in C#?**

* In C# the compiler won’t allow you to assign null value to a variable.
* C# 2.0 provide a special feature to allow us null value to a variable that is nullable type but only for the value type
* C# 8.0 provided us the feature to assign null values for the reference types also

**Explain the concept of boxing and unboxing.**

* **Boxing -** converting a value type to reference type
* **Un-boxing -** Converting a reference type to value type
* Boxing and unboxing are expensive operation because boxing involves heap allocation and the unboxing involves casting and type checking
* Avoid boxing and unboxing in performance critical application

**Explain ‘var’ keyword**

* ‘var’ is used to declare the implicitly typed variable, this means the type is determined at the compile time based on the initial value assigned to it.
* Once assigned the type cannot change
* It’s mostly used for declaring a cumbersome type variable and while using a LINQ query

**What is difference between const and readonly**

* Both are used to define a constant fields that are unchangeable. But the difference is where the value is assigned.
* ‘const’ - is a compile time constant. Whose value is initialized during the initialization. And the values cannot be changes.
* const is used where the value is known before the compilation and never changes
* ‘readonly’ - is a runtime constant, the value of readonly can be assigned while declaring or can be assigned in the constructor
* Readonly is used where the value is known at the runtime after compilation

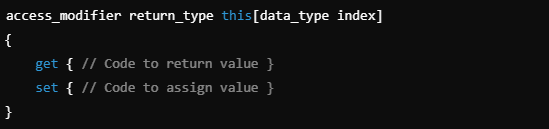
|  |  |
| --- | --- |
| **Const** | **readonly** |
| Value is assigned at declaration | Value is assigned at declaration or in constructor |
| Value cannot change | Value cannot change after assigned |
| Compile time field | Runtime field |

**What is properties in C#?**

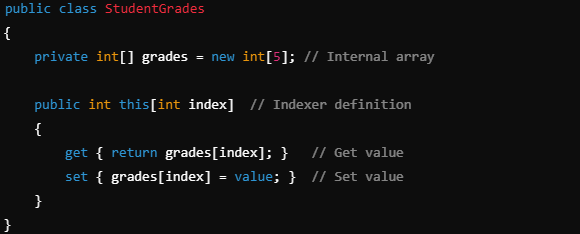
* Properties are special class member that is used to get and set the value of the field member safely, it allows controlled data access and modification
* Properties are used to where using for encapsulation (protected data access) , validataion and creating readonly or wirteonly variables.

**Explain the concept of Indexers?**

* Indexers are smart array the enables us to use the object of the class like array
* They provide a way to retrieve or set value of class or struct with the indexes

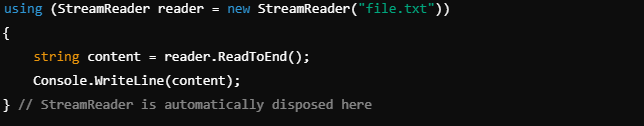


* Use ‘this’ keyword with the parameter inside []
* Use ‘get’ to retrieve value and ‘set’ to assign value



**Purpose of using keyword**

* ‘using’ can be used at two different context
* For managing the namespace
* For automatic resource management, it is used to release the resource of the object that implements the IDisposable Interface, by automatically calling the Dispose() method



**Explain the concept of Extension method**

* Extension method used to create additional functionality for the existing types without modifying the source code.
* They allow you to add methods to built-in types like string, int, List<T>, or even your custom classes.
* They are static methods but behave like instance methods of the type they extend.
* Syntax : they must be static and must be inside a static class and the first variable must use this before the type to extend