C#

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| I | | | | |
| 1.Introduction to C# | 1 | | | |
| 2.Reading and writing to Console | 2 | | | |
| 3.Built in types | 2 | | | |
| 4.Built in type C# | 3 | | | |
| 5.Common operator in C# | 3 | | | |
| 6.Nullable types in C# | 4 | | | |
| 7.Data type conversions in C# | 4 | | | |
| 8.Arrays | 5 | | | |
| 9.Comments | 5 | | | |
| 10.If statement | 5 | | | |
| 11.Switch statement | 5 | | | |
| 12.Switch statement(go to) | 6 | | | |
| 13.While loop | 7 | | | |
| 14.Do while loop | 8 | | | |
| 15.For and Foreach loops | 8 | | | |
| 16.Methods in C# | 8 | | | |
| 17.Method parameters | 8 | | | |
| 18.Namespaces | 10 | | | |
| II | | | | |
| 19.Introduction to class | 11 | | | |
| 20.Static and Instance class | 12 | | | |
| 21.Inheritance in C# | 13 | | | |
| 22.Method hiding in C# | 15 | | | |
| 23.Polymorphism | 16 | | | |
| 24.Method overriding and method hiding | 17 | | | |
| 25.Method overloading | 18 | | | |
| 26.Proprieties (get and set) | 19 | | | |
| III | | | | |
| 27.Proprieties in C# | | 21 | | |
| 28.Structs in C# | | 22 | | |
| 29.Diffrence between structs and classes | | 23 | | |
| 30.Interfaces | | 24 | | |
| 31.Explicit interface implementation | | 25 | | |
| 32.Abstract classes | | 27 | | |
| 33.Diffrence between abstract classs and interface | | 27 | | |
| 34.Problems of multiple class inheritance | | 28 | | |
| 35.Multiple class inheritance using interfaces | | 29 | | |
| 36.Delegates | | 30 | | |
| IV | | | | |
| 37.Delegates usage in C# | | 31 | | |
| 38.Delegates usage in C# -reusable code | | 32 | | |
| 39.Multicast Delegates in C# | | 33 | | |
| 40.Exception Handling -basics | | 35 | | |
| 41.Inner Exceptions | | 37 | | |
| 42.Custom Exceptions | | 38 | | |
| 43.Exception handling abuse | | 39 | | |
| 44.Preventing exception handling abuse | | 40 | | |
| V | | | | |
| 45.Enums introduction | | | 41 | |
| 46.Enums example | | | 41 | |
| 47.Enums definition and proprieties | | | 42 | |
| 48.Diffrence between Types and Type members | | | 44 | |
| 49.Acces modifiers | | | 45 | |
| 50.Acces modifiers – Internal and Internal Protected | | | 46 | |
| 51.Acces modifiers for types | | | 47 | |
| 52.Attributes | | | 47 | |
| 53.Reflection in C# | | | 49 | |
| 54.Reflection exemple | | | 51 | |
| 55.Late binding and early binding using reflection | | | 51 | |
| VI | | | | |
| 56.Generics | | | | 53 |
| 57.ToString method – how to override it | | | | 54 |
| 58.Why should we override Equals() method | | | | 55 |
| 59.Diffrence between ToString() and Convert.ToString() | | | | 56 |
| 60.Diffrence between System.String and System.Text.StringBuilder | | | | 56 |
| 61.Partial classes | | | | 57 |
| 62.Creating partial classes in C# | | | | 58 |
| 63.Partial methods | | | | 58 |
| 64.Indexers | | | | 59 |
| 65.Creating an indexer | | | | 60 |
| 66.Overloading indexers | | | | 61 |
| 67.Optional parameters | | | | 61 |
| 68.Making parameters in a method optional using method overloading | | | | 62 |
| 69.Making parameters method optional by specifying parameter default | | | | 62 |
| 70.Making parameters optional by using Optional Attribute | | | | 63 |
| 71.Code snippets in Visual Studio | | | | 63 |
| 72.What is a dictionary in C# | | | | 64 |
| V | | | | |
| 73.Dictionary in C# | | | | 65 |
| 74.List collection class in C# | | | | 66 |
| 75.List collection class – continue | | | | 67 |
| 76.Working with generic list class and ranges | | | | 67 |
| 77.Sort a list of simple types | | | | 68 |
| 78.Sort a list of complex types | | | | 69 |
| 79.Sorting a list of complex types using comparison delegate | | | | 70 |
| 80.List collection class – usefull methods | | | | 72 |
| 81.When to use a dictionary over a list in C# | | | | 73 |
| VI | | | | |
| 82.Generic Queue collection class | | | | 75 |
| 83.Generic stack collection class | | | | 77 |
| 84.Realtime example Queue | | | | 78 |
| 85.Realtime example Stack | | | | 79 |
| 86.Multithreading in C# | | | | 79 |
| 87.Advantages and disadvantages of multithreading | | | | 80 |
| 88.Thread start delegate | | | | 81 |
| 89.Parameterized Thread Start – delegate | | | | 83 |
| 90.Passing data to safe function in type safe maner | | | | 84 |
| VI | | | | |
| 91.Retriving data from thread function using callback method | | | | 85 |
| 92.Significance of Thread Join and Thread IsAlive functions | | | | 87 |
| 93.Protecting shared resources from concurrent access in multithreading | | | | 89 |
| 94.Diffrence between Monitor and lock in C# | | | | 91 |
| 95.Deadlocks | | | | 93 |
| VII | | | | |
| 96.How to resolve a deadlock in a multithread program | | | | 95 |
| 97.Performance of a multithreaded program | | | | 96 |
| 98.Anomymous Methods | | | | 97 |
| 99.Lambda Expressions | | | | 99 |
| 100.Func Delegate (Generic Delegate) | | | | 100 |