

C# Basics Quiz

From: Google Forms (forms-receipts-noreply@google.com)

To: deadrienhill@yahoo.com

Date: Friday, December 30, 2022, 05:19 PM CST

Google Forms

Thanks for filling out [C# Basics Quiz](#)

Here's what was received.

[View score](#)

C# Basics Quiz

This quiz will test your knowledge on C# basics. This is everything we have covered before SQL. Good Luck!

Email *

deadrienhill@yahoo.com

What type of Version Control System is git? *

- ☒ Distributed
- ☐ Centralized
- ☐ Intrinsic

Which of the following commands will create a new branch called new_work and also check it out?

- ☒ git checkout -b new_work
- ☐ git branch new_work
- ☐ git checkout new_work

In order to merge a new branch into master, you have to add and commit changes to that branch first *

- ☒ True
- ☐ False

Now we want to merge new_work into master. Which branch should we be on when we execute git merge new_work? *

- ☐ new_work
- ☒ master

If we wanted to add ALL modified files to the staging area, what git command would we use? *

- ☐ git add
- ☒ git add .
- ☐ git add <filename>

☐ git log

which git command will list the branches in your repository? *

- ☐ git checkout
- ☐ git commit
- ☒ git branch
- ☐ git branch <BranchName>

A _____ is the Git equivalent of a "save" *

- ☒ commit
- ☐ push
- ☐ remote

Which command pulls the latest version of the remote repository to our machine? *

- ☐ git push
- ☐ git commit
- ☐ git remote - v
- ☒ git pull

Which of the following commands will allow you to abort a merge? *

- ☒ git merge --abort

- ☐ git abort
- ☐ git merge --abort --conflict

The command _____ is a combination of fetch and merge. *

- ☒ git pull
- ☐ git commit
- ☐ none of the above

_____ enables you to compare changes in the working directory against a previously committed version. *

- ☒ git diff
- ☐ git compare
- ☐ git previous
- ☐ none of the above

_____ files contain all the source code that are compiled. It also contains compiler settings and other configuration files *

- ☒ project
- ☐ solution
- ☐ source code file

This command creates a new console project for us: *

- ☒ dotnet new console
- ☐ dotnet new
- ☐ dotnet console
- ☐ mkdir console

This command is used to execute unit tests in a given project *

- ☒ dotnet test
- ☐ dotnet run test

All external dependencies used in your app are stored in what folder? *

Select All That Apply

- ☒ Packages
- ☒ Dependencies
- ☐ DLC Folder
- ☐ Nuget

What is the most popular package manager for .NET development? *

- ☐ DLC
- ☒ Nuget
- ☐ OpenPKG
- ☐ Zero Install

While you're debugging, you can use what window to track variables and expressions? *

- ☐ Solution explorer
- ☒ Watch window
- ☐ Intellisense
- ☐ Call Stack
- ☐ Task List

Where can you view the function or procedure calls that are currently on the stack? *

- ☐ Solution explorer
- ☐ Watch window
- ☐ Task list
- ☒ Call stack

True or False: The debugger can break execution at the point where an exception is thrown, so you may examine the exception before a handler is invoked *

- ☒ True
- ☐ False

This is a storage location used by computer memory to store data for usage by an application *

- ☐ Constants
- ☒ Variables
- ☐ Datatypes

Which is a valid C# variable name? *

- ☐ 1car
- ☐ #LastStudent#
- ☒ wrongOne

Which of the following is a correct example of how to initialize a variable *

- ☐ int = 42;
- ☐ STRING myString = "Hello!";
- ☐ int 1number = 1;
- ☒ bool isParsed = false;

What are the 3 parts to a variable? *

- ☐ name, value, semicolon
- ☒ datatype, name, value
- ☐ public, static, int
- ☐ datatype, static, value

True/False. You can use a variable before it is declared. *

- ☐ True
- ☒ False

C# is strongly-typed. *

- ☒ True
- ☐ False

Choose the correct definition of strongly typed *

- ☒ Once an object has a type, that type will NEVER change
- ☐ Every object MUST have a type before the program will compile

Choose the correct definition of statically typed *

- ☐ Once an object has a type, that type will NEVER change
- ☒ Every object MUST have a type before the program will compile

To initialize a variable, you do not need to assign it a value. *

- ☐ True
- ☒ False

To declare a variable, you do not need to assign it a value. *

- ☒ True
- ☐ False

_____ types in C# automatically support being set null *

- ☐ Value
- ☒ Reference

The _____ is a FIRST IN, FIRST OUT collection of elements *

- ☐ stack
- ☒ queue
- ☐ heap

The _____ utilizes a LAST IN, FIRST OUT procedure *

- ☒ stack
- ☐ heap
- ☐ queue

_____ types do NOT automatically support being set to null and require an actual value *

- ☒ Value
- ☐ Reference

These types are stored in memory in a location called the stack *

- ☒ value
- ☐ reference

A value type, followed by a ? is shorthand syntax for _____ *

- ☒ nullable
- ☐ expendable
- ☐ this is not allowed
- ☐ intrinsic

Of the types below, select all that are reference types: *

- ☐ int
- ☒ string
- ☒ array
- ☐ char
- ☐ enum

Every C# program has this method: *

- ☒ Main
- ☐ Program

- ☐ Start
- ☐ Console

Do methods have to return information *

- ☒ No
- ☐ Yes

in "public bool SendEmail(string emailAddress)" what is "(string emailAddress)" *

- ☐ Attribute
- ☐ Return Type
- ☐ Modifier
- ☒ Parameter

Optional keywords that give certain qualities to a method *

- ☐ return type
- ☐ privacy attributes
- ☐ parameters
- ☒ modifiers

Which operator is used to determine the remainder? *

- ☐ *

☐ /☒ %

Operators that require two operands are called: *

☐ Unary Operator☐ Ternary Operator☒ Binary Operator

(a += b) is the same as: *

☒ a = a + b☐ a = (a+=b)☐ a == b

What is the output of this code? *

```
using System;

namespace SpringClean
{
    class Program
    {
        static void Main(string[] args)
        {
            int kidToysToKeep;
            int kidToysToDonate;

            kidToysToDonate = 12;
            kidToysToKeep = 7;

            int kidToysToBeginWith = kidToysToKeep + kidToysToDonate;

            Console.WriteLine(kidToysToBeginWith);
        }
    }
}
```

- ☐ "kidsToysToBeginWith"
- ☐ 12
- ☒ 19

'=' means equal to. *

- ☐ True
- ☒ False

5 >= 5 returns true. *

- ☒ True
- ☐ False

40 <= 39 evaluates to: *

- ☐ True
- ☒ False

What does CCR stand for? *

- ☐ Clean, Concise, Readable
- ☐ Concise, Clean, Readable
- ☒ Clear, Concise, Readable
- ☐ Concise, Clear, Readable

C# was developed by: *

- ☐ Apple
- ☐ Google
- ☒ Microsoft
- ☐ IBM
- ☐ Amazon

Type Inference uses which syntax? *

- ☐ string
- ☐ bool
- ☒ var

☐ object

C# is a: *

- ☒ Strongly-Typed Language
- ☐ Weakly-Typed Language
- ☐ Dynamically-Typed Language
- ☐ Syntax-Typed Language

What character allows us to use string interpolation? *

- ☐ !
- ☒ \$
- ☐ %
- ☐ :
- ☐ ()

An "if" statement must contain an "else". *

- ☐ True
- ☒ False

Will the scope of this "if" statement execute? *

if (5 > 10 || "Hey" == "Hey") { /* some code */ }





Yes



No



Unable to be determined

Will the scope of this "if" statement execute? *

```
if (myInt > 10 && "Hey" == "Hey") { /* some code */ }
```



Yes



No



Unable to be determined

Why do we use Switch Statements instead of else-if statements? *



else-if statements can only be used for 2 conditions.



switch statements increase readability and are more optimized for a larger number of conditions.



else-if and switch statements are exactly the same, so it doesn't matter.

C# is case sensitive. *



true



false

In a switch statement, multiple default statements are allowed. *



True

☒ False

What are the 4 Iteration Statements we've discussed in class? *

- ☒ for
- ☐ do
- ☒ do-while
- ☐ while-do
- ☒ foreach
- ☒ while

True or False: foreach can only be used with a collection? *

- ☒ True
- ☐ False

What is the maximum number of times a loop can execute? *

- ☐ 2
- ☐ 5
- ☐ 7
- ☐ 1
- ☒ Infinitely or until there is no more space on the hard drive

Which part of this for loop is the Conditional? *

```
for ( int i = 0; i < 13; i++) { Console.WriteLine(numbers[i]); }
```

- ☐ for (
- ☐ int i = 0;
- ☒ i < 13;
- ☐ i++;)
- ☐ { Console.WriteLine(numbers[i]); }



Thank you

What is the namespace needed to use List<T>? *

- ☐ System.List.Generic
- ☐ System.Collections
- ☐ System.List
- ☒ System.Collections.Generic

Which of these is used to identify the amount of elements in a List<T>? *

- ☐ .Length
- ☒ .Count
- ☐ .Max
- ☐ .MaxNumber

How many elements are in this array? *

```
int[] weekDays = new int[7];
```

- ☐ 4
- ☐ 6
- ☐ 3
- ☒ 7

True or False: To use Arrays, you first need to use the System.Collections.Generic; namespace *

- ☐ True
- ☒ False

Which of these is used to identify the number of elements in an Array? *

- ☐ .Count
- ☐ .Max
- ☒ .Length
- ☐ .MaxNumber

Method overloading is when multiple methods have the same _____, but different _____.

- ☐ parameters, name
- ☐ syntax, variables
- ☒ name, parameters

True or False: This won't compile because it has the same signature: *

```
static int Multiply(int a, int b)
{
    return a * b;
}

static int Multiply(int c, int d)
{
    return c * d;
}
```

- ☒ True
- ☐ False

When overloading methods, the definitions of the methods must differ from each other by the parameter types and/or number of parameters *

- ☒ True
- ☐ False

What are the 3 ****Required**** parts of a Class? *

- ☐ Static
- ☒ Name
- ☒ Class Keyword
- ☐ Parameters
- ☒ Scope

Which of the following are items you may find IN a Class? *

- ☒ Methods
- ☐ Namespace
- ☒ Properties
- ☒ Constructors

What part of this class instantiation is circled in red? *

```
Dog myFirstDog = new Dog();
```

- ☐ Name
- ☐ Type
- ☐ Property
- ☒ Constructor
- ☐ Field

You can only have 1 constructor per class? *

- ☐ True
- ☒ False

Which of the following is a correctly created class? *

```
Dog class
{
    private int _numberOfLegs;
    public int Age;
    public string Name { get; set; }
}
```

☐ Option 1

```
public Dog()
{
    private int _numberOfLegs;
    public int Age;
    public string Name { get; set; }
}
```

☐ Option 2

```
class Dog
{
    private int _numberOfLegs;
    public int Age;
    public string Name { get; set; }
}
```

☒ Option 3

```
new class Dog
{
    private int _numberOfLegs;
    public int Age;
    public string Name { get; set; }
}
```

☐ Option 4

What symbol do you use to access class members? *

☐ ;

☐ ,

☐ " "

☒ .

☐ ()

Which of the following demonstrates object initializer syntax? *


```
Dog d = new Dog ()  
{  
    int Age = 5;  
    string Name = "Spot";  
}
```

☐ Option 1

```
Dog d = new Dog ();  
d.Age = 5;  
d.Name = "Spot";
```

☐ Option 2

```
Dog d = new Dog  
(  
    Age = 5,  
    Name = "Spot"  
);
```

☐ Option 3

```
Dog d = new Dog()  
{  
    Age = 5,  
    Name = "Spot"  
};
```

☒ Option 4

How many class members does this class have? *

```
class Dog
{
    private int _numberOfLegs;
    public int Age { get; set; }
    public string Name { get; set; }

    public Dog(string _name)
    {
        Name = _name;
    }

    public void Bark()
    {
        //do something
    }

    public void WagTail()
    {
        //code goes here
    }
}
```

- ☐ 2
- ☐ 1
- ☒ 5
- ☐ 3

Creating a class file is the same as instantiating a class? *

- ☐ True
- ☒ False

What is the default access modifier for a class member? *

- ☐ internal
- ☒ private
- ☐ public
- ☐ protected

Constructors are special types of methods of a class that get executed when its object is created *

- ☒ True
- ☐ False

Can we define a method with the same class name in C#? *

- ☐ Yes
- ☒ No
- ☐ Maybe

How many types of constructors are there in C#? *

- ☐ 10
- ☐ 9
- ☐ 20



5

Constructor without a parameter is called *

- ☐ static constructor
- ☐ Null Constructor
- ☒ Default Constructor

How many constructors can be defined in a class? *

- ☐ 9000
- ☒ An optional no-argument constructor plus 'n' number of parameterized constructors.
- ☐ One no-argument constructor

We cannot create instances of static classes. Can we have constructors for static classes? *

- ☒ Yes
- ☐ No

Can you prevent a class from being instantiated? *

- ☒ Yes
- ☐ No

*



☒ Option 1

What is the syntax for inheritance? *

- ☐ semicolon ;
- ☒ colon :
- ☐ dot .

☐ comma ,

Which class is the base class? *

class Truck : Vehicle

- ☐ class
- ☐ Truck
- ☒ Vehicle

True or False : The derived class implicitly gains all the members of the base class

*

- ☒ True
- ☐ False

True or False: Constructors are inherited *

- ☐ True
- ☒ False

Which class is the derived class? *

class AI : Human

- ☐ class
- ☐ Human
- ☒ AI

How many classes can a single class inherit from? *

- ☒ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ Infinite

Inheritance allows the _____ class to reuse the code in the _____ class without having to rewrite it. *

- ☒ derived , base
- ☐ base, derived

Which of the following statements are true about static classes? *

- ☐ Static classes CAN be instantiated
- ☒ Static classes CANNOT be instantiated
- ☐ Static classes CAN have non-static members
- ☒ Another class CANNOT inherit from a static class

Which of the following is an example of a properly created static class? *


```
class Dog
{
    public static string Name { get; set; }
    public static int Age { get; set; }

    public static void Bark()
    {
        Console.WriteLine("WOOF");
    }
}
```

☐ Option 1

```
class static Dog
{
    public static string Name { get; set; }
    public static int Age { get; set; }

    public static void Bark()
    {
        Console.WriteLine("WOOF");
    }
}
```

☐ Option 2

```
static class Dog
{
    public static string Name { get; set; }
    public static int Age { get; set; }

    public static void Bark()
    {
        Console.WriteLine("WOOF");
    }
}
```

☒ Option 3

```
class Dog : static
{
    public static string Name { get; set; }
    public static int Age { get; set; }

    public static void Bark()
    {
        Console.WriteLine("WOOF");
    }
}
```

☐ Option 4

How many times does the constructor get called for a static class? *

- ☐ 2 times
- ☒ 1 time
- ☐ 0 times (it never gets called)

Which of the following demonstrates how you would invoke a static method *

```
new Dog.Bark();
```

☐ Option 1

```
Dog pupper = new Dog();  
pupper.Bark();
```

☐ Option 2

```
Dog.Bark();
```

☒ Option 3

```
static Dog.Bark();
```

☐ Option 4

How many copies of static members are there per class? *

- ☐ 2
- ☒ 1
- ☐ undefined

Only the methods in a static class need to be made static *

- ☐ True
- ☒ False

Why would you use an abstract class? *

- ☐ To enable multiple inheritance
- ☒ To create a partial class that serves as a template for its derived classes
- ☐ When you want have a base class to instantiate

Which of the following describes Abstract Classes? *

- ☐ It is an example of static polymorphism
- ☒ It is an example of dynamic polymorphism

Which of the following describes Polymorphism *

- ☐ Combining similar properties and characteristics into a common object

- ☐ Information hiding
- ☒ One thing having many forms

What is dynamic polymorphism? *

- ☒ Objects of a derived class may be treated as objects of a base class
- ☐ Method overloading

When would you want to use the abstract keyword in an abstract class? *

- ☐ When you want the derived class to define the method implementation or accessor
- ☒ When you want the abstract class to define the method or accessor
- ☐ When you want the abstract class to define the method or accessor and give the derived class the ability to override the implementation

When would you want to use the virtual keyword in an abstract class? *

- ☐ When you want the derived class to define the method or accessor
- ☐ When you want the abstract class to define the method or accessor
- ☒ When you want the abstract class to define the method or accessor and give the derived class the ability to override the implementation

What is an abstract class? *

- ☐ An abstract class is one without any child classes
- ☐ An abstract class is any parent class with more than one child class

- ☒ An abstract class is class which cannot be instantiated, but can be a base class
- ☐ abstract class is another name for "base class"

What is an advantage of polymorphism? *

- ☒ The same program logic can be used with objects of several related types
- ☐ Variables can be re-used in order to save memory
- ☐ Constructing new objects from old objects of a similar type saves time
- ☐ Polymorphism is a dangerous aspect of inheritance and should be avoided

An interface does not care about the _____. It merely requires that it is implemented.

- ☐ Base class
- ☒ Implementation
- ☐ Properties
- ☐ Methods

What is the correct Interface naming convention of the following:

- ☐ IsampleInterface
- ☐ IOtherinterface
- ☒ ISampleInterface

True or False: Classes can implement multiple interfaces? *

- ☒ True
- ☐ False

What is the syntax to implement an interface? *

- ☐ semicolon ;
- ☒ colon :
- ☐ comma ,
- ☐ dot .

How many times can you implement the same Interface? *

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☒ As many times as you have classes

True or False: You must use an interface if you want to simulate inheritance for structs *

- ☒ True
- ☐ False

When a class or struct implements an interface, the class or struct must provide the _____.*

- ☐ Base Class
- ☒ Implementation
- ☐ Abstraction

Factory Pattern is: *

- ☐ A design pattern that ensures only one instance of a given object exists
- ☐ A design pattern in which one class copies other instances of objects
- ☒ A design pattern that utilizes one class to handle object creation at Runtime
- ☐ A design pattern where an object bridges the gap between two objects

Factory Pattern can only be achieved by using an interface? *

- ☐ True
- ☒ False

Does factory pattern utilize polymorphism? *

- ☒ Yes
- ☐ No

When the client passes the information to the program to create the class, where is the class instantiated? *

- ☐ In the concrete class
- ☒ In the factory
- ☐ In the interface/abstract class
- ☐ On the client side



D'oh!

What does LINQ stand for? *

- ☐ Language Interpreted Query
- ☐ Linq Integrated Query

- ☒ Language Integrated Query
- ☐ Liskovs Inversion Query

Select the two types of syntax used in LINQ *

- ☒ Query Syntax
- ☐ LINQ Syntax
- ☐ SQL Syntax
- ☒ Method Syntax

What namespace is needed to access LINQ? *

- ☐ System.Collection.Linq
- ☐ System.Collection.Generic
- ☐ Linq.Collection
- ☒ System.Linq

What does the Where() method do? *

- ☐ Gets a single element from the sequence
- ☒ Filters a sequence of values based on a predicate.

What concept does Generics bring to the .NET Framework? *

- ☐ System.Collections.Generic;

- ☒ Type Parameters
- ☐ ArrayLists

True or False: Constraints inform the compiler about the capabilities a type argument must have. *

- ☒ True
- ☐ False

Where T : class = *

Match to the definition

- ☐ The type argument must be a non-nullable value type.
- ☒ The type argument must be a reference type
- ☐ The type argument must be a non-nullable type.
- ☐ The type argument must be or implement the specified interface.

True or False: The elements in a list cannot be inserted and removed dynamically.

*

- ☐ True
- ☒ False

What are the 3 types of errors? *

- ☐ Debugging Error

- ☒ Compile Time Error
- ☐ Interface Error
- ☒ Runtime Error
- ☒ Logical Error

What is an Exception? *

- ☐ An error that will only show up at compile-time
- ☒ An error that occurs during runtime of an application

Will this line of code throw an exception? *

```
int i = 10 / 0;
```

- ☒ Yes
- ☐ No
- ☐ Not enough information

Will this cause an error? *

```
int[] numbers = new int[5] { 11, 32, 43, 65, 3 };  
Console.WriteLine(numbers[5]);
```

- ☒ Yes
- ☐ No
- ☐ Not enough information

Testing your app is a good way to prevent debugging altogether *

- ☒ True
- ☐ False

True or False: A try block is used by C# programmers to partition code that might be affected by an exception. *

- ☒ True
- ☐ False

How many catch blocks can a try block have with it? *

- ☐ No more than 3
- ☐ 2 or more
- ☒ 1 or more
- ☐ none

True or False: A finally block does not always run *

- ☐ True
- ☒ False

Using Exception in the exception filter of a catch is considered to be the most specific and thus should be the first catch block? *

- ☐ True
- ☒ False

Using TDD (Test Driven Development) what is the first step after writing just enough code to complete the test? *

- ☒ Run the Test and watch it fail
- ☐ Implement the feature

This is where we actually call the method you want to test *

- ☒ Act
- ☐ Arrange
- ☐ Assert

A _____ is the smallest testable part of any software *

- ☐ method
- ☒ unit
- ☐ test

This is where we check against a constant (check against what is expected) *

- ☐ Arrange
- ☐ Act

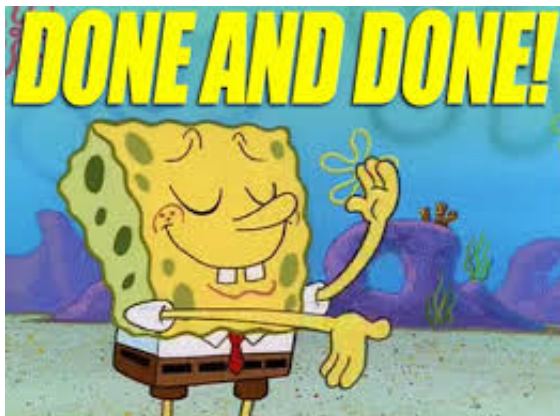
☒ Assert

When we change code without changing how the code behaves *

- ☐ instantiation
- ☐ inheritance
- ☒ refactoring
- ☐ initialization

This is where we prepare the code in order to call the method you need to test *

- ☐ Act
- ☒ Arrange
- ☐ Assert



☒ Option 1

[Create your own Google Form](#)
[Report Abuse](#)