Q1. What is false about Interfaces in C#?

* They can be used to allow for multiple inheritance
* All methods in an interface must be implemented
* Both a class and a struct can implement them
* You define an interface with the interface keyword
* **All of these are true about interfaces**

Q2. What is a using block in C# used for?

* To import namespaces
* To add references to your class
* **To dispose unmanaged resources**
* To handle exceptions with a try/finally structure

Q3. Is C# a strongly typed language, what is a strongly typed language?

* **Yes, each type of data is predefined as part of the programming language**
* Yes, data types do not have to be explicitly called
* No, each type of data is predefined as part of the programming language
* No, data types do not have to be explicitly called

Q4. In C#, one class can inherit from multiple base classes. TRUE/**FALSE**

Q5. Which of the following is the correct operator to compare two variables?

* :=
* =
* Equal
* **== (answer)**

Q6. Which of the following is not a valid data type?

* Float
* **Real**
* Int
* Double

Q7. The first index number in an array starts with \_\_\_\_ and the highest index number of an array of size n will be \_\_\_\_\_.

* **0, n-1**
* 1, n-1
* 0, n
* 1, n

Q8. Which of the following two entities (reading from Left to Right) can be connected by the dot operator?

* A class member and a class object
* A class object and a class
* A class and a member of that class
* **A class object and a member of that class.**

Q9. What is garbage collection?

* The process of negating methods from a super class
* **The process of de-allocating memory automatically**
* The process of restoring orphaned objects.
* The process of creating a new object

Q10. What is true about a class?

* It should be private always.
* It is blueprint for a schema
* **It may contain any number of variables and methods.**
* It should contain at least one variable and at least one method.

Q11. An If-Else statement is used for handling thrown Exceptions

* True
* **False**

Q12. Which of the following are the correct ways to increment the value of a variable by 1?

* ++a++;
* **a+= 1;**
* a ++ 1;
* **a = a+1;**
* a = +1;

Q13. Which of the following is NOT an Arithmetic operator in C#.Net?

* **\*\* (answer)**
* +
* /
* %
* \*

Q14. Which of the following statements are correct about static functions?

* **Static functions can access only static data.**
* **Static functions cannot call instance functions.**
* It is necessary to initialize static data.
* **Instance functions can call static functions and access static data.**
* *this* reference is passed to static functions

Q15. Which of the following can be facilitated by the Inheritance mechanism?

* **Use the existing functionality of base class**
* **Override the existing functionality of base class.**
* **Implement new functionality in the derived class.**
* Implement polymorphic behavior.
* Implement containership

Q16. Which of the following statements are correct about the C#.NET code snippet given below?

namespace IndiabixConsoleApplication

{

class index

{

protected int count;

public index()

{

count = 0;

}

}

class index1: index

{

public void increment()

{

count = count +1;

}

}

class MyProgram

{

static void Main(string[] args)

{

index1 i = new index1();

i.increment();

}

}

}

* *count*  should be declared as public if it is to become available in the inheritance chain.
* ***count* should be declared as protected if it is to become available in the inheritance chain.**
* **While constructing an object referred to by *i* firstly constructor of index class will be called followed by constructor of *index1* class.**
* **Constructor of index class does not get inherited in *index1* class.**
* *count* should be declared as Friend if it is to become available in the inheritance chain.

Q17. What will be the size of the object created by the following C#.NET code snippet? **24Bytes**

namespace IndiabixConsoleApplication

{

class Baseclass

{

private int i;

protected int j;

public int k;

}

class Derived: Baseclass

{

private int x;

protected int y;

public int z;

}

class MyProgram

{

static void Main (string[ ] args)

{

Derived d = new Derived();

}

}

}

Q18. Which of the following statements are correct about data types?

* **If the integer literal exceeds the range of byte, a compilation error will occur.**
* **We cannot implicitly convert non-literal numeric types of larger storage size to byte.**
* Byte cannot be implicitly converted to float
* A char can be implicitly converted to only int data type.
* **We can cast the integral character codes.**

Q19. Which of the following is an 8-byte Integer?

* Char
* **Long**
* Short
* Byte
* Integer

Q20. Which of the following control statements is incorrect?

* if (condition) else if {//some code}
* if(condition1) else if {//some code} else {//some code}
* if(condition) else {//some code}
* **if(condition1) else {//some code} else if(condition2) {//some code}**

Q21. How do you access a private field that is a member of a class?

* **With a public properly to get and set the private field.**
* Use the dot ‘.’ Operator to access the field like other class members.
* It is not possible.
* With a private property to get and set the private field

Q22. Used in a class definition, what is the purpose of the protected access modifier?

* **To hide access to the field from outside the class except a derived class.**
* To ensure a class field is read-only from outside the class.
* To ensure class field is read-only from outside the class except a derived class.
* To hide access to the field from outside the class by all

Q23. Which of the following sets are assignment operators?

* **\*-, +=, =, /= (answer)**
* **\*=, ==, !=, +=**
* **\*=, =, >=, +=**
* **\*=, ==, <=. -=**

Q24. In a switch statement, if the value of the expression is not represented by a case what will happen?

* **The default case will execute if provided in the switch statement**
* Nothing, program execution continues to the statement after the switch
* The program will throw an error and crash
* The last case is executed

Q25. In a literal string, which character is used to start an escape sequence?

* **\ (answer)**
* @
* !
* %

Q26. What is the output for the following code?

  bool b1 = 1 < 0;  
  bool b2 = !(12 % 3 > 2);  
  if (b1 = b2)  
    Console.Write("true: ");  
  else  
    Console.Write("false: ");  
  Console.WriteLine("b1 is {0}.  b2 is {1}.", b1, b2);

* true: b1 is False. b2 is False
* false: b1 is False. b2 is True.
* false: b1 is False. b2 is False.
* **true: b1 is True. b2 is True.**

Q27. What is the output for the following code?

  int i = 9;  
  double j = 0.2;  
  Console.WriteLine(++i/j-i++);

* 35
* 36
* 39
* **40**

Q28. Which of the following is not a value type?

* Double
* Char
* Integer
* **Array**

Q29. Which loop prints the elements of an integer array called arr?

* while (int x=0; x<arr; x++) { Console.WriteLine(x);}
* for (int x=0; x<arr; x++) {Console.WriteLine(x);}
* **foreach (int x in arr) {Console.WriteLine(x);}**
* do { Console.WriteLine(x) } while (int x = 0; x<arr; x++);

Q30. Given a class Robot, which is the correct way to instantiate an object?

* **Robot droid = new Robot();**
* Robot droid = Robot();
* Droid = new Robot();
* Robot droid;

Q31. What is the correct output of the following code sample?

    namespace eIntern {       
    class Robot {  
        string type;  
        double speed;  
        public void SetRobot(double speed) {  
            type = "hover";  
            speed = speed;  
        }  
        public void Output() {  
            Console.WriteLine(type +"bot has speed " + speed);  
        }  
    }  
    class Program {  
        static void Main(string[] args) {  
            Robot droid = new Robot();  
            droid.SetRobot(32);  
            droid.Output();  
        }  
    }  
}

* **hoverbot has speed 0**
* hoverbot has speed 32
* hoverbot has speed 32.0
* Error will prevent code from executing.

Q32. Within a class definition, what is the default access modifier for a class member without one?

* **private**
* public
* protected
* internal

Q33. What value is returned by the following method signature? Protected void MyMethod();

* **None.**
* Null.
* 0
* True

Q34. For a class definition with 2 fields, how many parameters does a default constructor have?

* 2
* 3
* **0**
* 1

Q35. Which of the following is an incorrect signature for a constructor for a Robot class?

* **public void Robot(int speed)**
* public Robot()
* public Robot(int speed)
* public Robot(int speed, string type)

Q36. Given an ArrayList arList, how do you get the number of elements in arList?

* **arList.Count**
* arList.Length
* arList.MaxIndex
* arList.Capacity

Q37. For a stack type of collection, which of the following is a valid method?

* **Pop**
* Insert
* Add
* Append

Q38. At most, how many times can a value be repeated in a Dictionary collection?

* **Unlimited.**
* It must be unique
* 2
* 3

Q39. What is the correct way to declare and instantiate a List collection?

* **List<int> numbers = new List<int>();**
* List numbers = new List();
* List[] numbers = new List[int]();
* List <> numbers = new List<int>();

Q40. Of the following collections, which is not considered dynamic?

* **string[]**
* List
* Disctionary
* ArrayList

Q41. Provided a List with 10 names, which is the correct way to get the name at the 5th position in the List?

* **names[4];**
* names.ElementAt(5);
* names.ItemAt(5);
* names.AtIndex(4);

Q42. Given the following class and interface signatures, which is the best way to apply multiple inheritance to DerivedRobot class?

* **class Robot1 : Robot A {//code} class DerivedRobot : RobotA, Irobot2, IrobotB {//code}**
* class DerivedRobot: Robot1, RobotA, Irobot2, IrobotB {//code}
* class Robot1 : Irobot2 {//code} class RobotA : IrobotB {//code} class DerivedRobot : Robot1, RobotA {//code}
* class Robot1 : IrobotB, RobotA {//code}

Q43. Which is the base exception class that can catch all types of exceptions?

* **Exception**
* Base Exception
* GenericException
* ExceptionBase

Q44. What is the output after executing the code below which attempts to divide by zero?

            int i = 1;  
            int x = 0;  
            try {  
                int divideByZeroError = i / x;  
            }  
            catch (NullReferenceException ex) {  
                Console.WriteLine("Caught a null error");  
            }  
            catch (ArithmeticException ex) {  
                Console.WriteLine("Caught a math error");  
            }  
            catch (DivideByZeroException ex) {  
                Console.WriteLine("Caught a division error");  
            }

* **Caught a math error**
* Caught a division error
* Caught a null error
* None, the program terminates.

Q45. Like the default clause in a Switch statement executes for expressions without a case, will the finally clause catch the uncaught error in a try-catch statement?

* **No.**
* Yes.
* Only if specified with the base Exception class.

Q46. Which key word is used to add a base class library or namespace to a project to become accessible?

* **using**
* import
* include
* append

Q47. How would you decorate a class definition to make it nonconcrete?

* **abstract**
* virtual
* override
* sealed

Q48. Is possible to prevent a class from being inherited?

* **Yes, make it sealed.**
* Yes, make it nonconcrete.
* Yes, make it private.
* No.

Q49. Of the following components, which cannot be inside a class definition?

* **namespace**
* method
* field
* property

Q50. Which operator is used in order to increment a number by 1?

* **++ (answer)**
* --
* &&
* +

Q51. The modulus ‘%’ returns the remainder of two int values when divided. **TRUE**

Q52. You can use access modifiers in an interface. **FALSE**

Q53. Downcasting is implicit. **FALSE**

Q54. What are Assemblies?

* **Packages containing MSIL instructions and metadata to allow various components and tools to function together within the CLR at runtime.**
* A virtual environment that sits on top of an OS to convert the MSIL code into executable machine code at runtime.
* Commonly used functionality shared by all languages and .NET programs
* Supervises resource usage and automatically allocates and releases memory requirements

Q55. What do you use as a flag or marker indicating a place to pause current execution of a program?

* **Breakpoint**
* Step Into (Explanatin: Step Into is used after a breakpoint is set)
* Debug Mode
* Build

Q56. A class can have no constructors. **FALSE** (Note: A class will always have at least an implicit default constructor)

Q57. Which keyword is used to called a method from the Parent class?

* **base**
* this (Explanation: this keyword is used to call a class member from the Child class)
* virtual (Explanation: virtual keyword is used to specify a method that can be overridden in a derived class)
* override (Explanation: override keyword is used to override a method from the base case)

Q58. What will you use in order to mimic multiple inheritance?

* Abstract Class (Explanation: You can only inherit from one abstract class)
* **Interfaces**
* Virtual Classes (Explanation: The virtual keyword is used on methods and properties)
* override (Explanation: override keyword is used to override a method from the base class)

Q59. You can only have one constructor. **FALSE**

Q60. Properties are used to store private backing fields. **TRUE**

Q61. In C#, which project templates involve a Main method?

* **console app**
* class library
* xUnit test project

Q62. The garbage collector allocates a \_\_\_ to store and manage objects.

* Stack trace
* CTS
* **Managed Heap**
* Large Object

Q63. In C#, constructors have to initialize ever field. **FALSE**

Q64. In .NET. the VES is the execution environment for \_\_\_\_ code.

* reference-type
* Windows-specific
* **managed**
* unmanaged

Q65. Which component manages code execution?

* **Common Language Runtime**
* .NET Framework
* Common Language Specification
* Base Class Library

Q66. Which are .NET implementation following the CLI?

* **Mono**
* CoreFx
* **.NET Core**
* .NET Standard
* **.NET Framework**