1) Identify the Reference types ?

(Choose two correct options)

a) string

b) delegate

c) enum

d) int

Ans: string, delegate

2) What will the output for the following code ?

```
class SampleClass
  {
     public static int GetAdd(int num1,int num2)
     {
        Console.WriteLine("Addition is {0}",(num1+num2));
        return (num1 + num2) / 2;
     }
     public static int GetMul(int Val1,int Val2)
     {
        Console.WriteLine("Multiplication is {0}",(Val1*Val2));
        return (Val1 * Val2) / 2;
     }
  }
  class Program
  {
     delegate int TestDelegate(int n1, int n2);
     static void Main(string[] args)
     {
        TestDelegate testDelegate = null;
```

```
        testDelegate += SampleClass.GetAdd;
        testDelegate += SampleClass.GetMul;
        Console.WriteLine("Final Result {0}",testDelegate(4,4));
        Console.ReadLine();
    }
  }
```

Ans: Addition is 8

Multiplication is 16

Final Result 8

3) What will be the output for the following program?

```
using System;
class Program
  {
    static void Main(string[] args)
    {
      string fruit1 = "Banana";
      string fruit2 = fruit1.Replace('a', 'e');
      Console.WriteLine(fruit2);
      Console.ReadLine();
    }
  }
```

Ans : Benene

4) What will be the output for the following program?

```
using System;
class DynamicDemo
  {
```

```
    static void Main(string[] args)
    {
        dynamic val1 = 500;
        dynamic val2 = "jyothi";
        val2 = val1;
        Console.WriteLine("val1={0},val2={1}",val1.GetType(),val2.GetType());
        Console.ReadLine();
    }
}
```

Ans : val1=System.Int32,val2=System.Int32

5) The class which doesn't allow objects creation, but represents as parent class of child classes, is known as _____

a) Static

b) Sealed

c) Abstract

d) Partial

Ans: Abstract

6) Which of the following statements are TRUE about private Assembly?

a. Application which refers private assembly will have its own copy of the assembly

b. There will be only one copy of the assembly and it will be stored in Global assembly cache

options:

a) only a

b) only b

c) Both a and b

d) Neither a nor b

Ans :Both a and b

7) What will be the output for the following program?

NOTE : Line numbers are only for reference

```
class Program {
   static void Method(int[] num)//line1
      {
         num[0] = 11;
      }
      static void Main()
      {
         int[] numbers = { 1, 2, 3 };
         Method(numbers);//line2
         Console.WriteLine(numbers[0]);
      }
   }
```

Ans : 11

8) What will be the output for the following program?

```
using System;
class A
{
   public A()
   {
      Console.Write("Class A" + " ");
   }
}
class B : A
{
   public B()
   {
```

```
        Console.Write("Class B" + " ");
    }
}
class ConstructorChainDemo
{
    public static void Main(string[] args)
    {
        A bobj = new A();
    }
}
```

options:

a) class A class B

b) class B class A

c) class B

d) class A

Ans: class A

9) Consider the following languages specifications which must be met by a programming component to be re-used across multiple languages:

Instance members must be accessed only with the help of objects

The above specifications is provided by ?

a) Common Language Specifications(CLS)

b) Common Type System(CTS)

c) Attributes

d) JIT Compiler

Ans: CLS

10) Method overloading is a concept related to?

[Choose the most appropriate option]

a) dynamic polymorphism

b) static polymorphism

c) abstract class

d) encapsulation

Ans: static polymorphism

11) What will be the output for the following program?

```
using System;
class StaticDemo
{
   private static int number = 100;
   static StaticDemo()
   {
      number = number + 1;
   }
   public StaticDemo()
   {
      number = number + 1;
      Console.Write(number + " " );
   }
}
class NormalConstructionProgram
{
   static void Main(string[] args)
   {
      StaticDemo obj= new StaticDemo();
   }
}
```

options:

a) 100

b) 101

c) 102

d) 103

Ans: 102

12) Which of the following statements are TRUE about generics?

[Choose two correct options]

a) Generics does not need to perform boxing

b) Generics does not need to perform unboxing

c) Explicit typecasting is required in generics

d) a generic declared to hold integer values can hold both integer and string values

Ans: Generics does not need to perform boxing

13) _____ class is used to find out object's metadata i.e, methods, fields, properties at

a) System.Type

b) System.Reflection

c) System.Assembly

d) System.String

Ans: System.Type

14) What will be the output for the following program?

```
class Test
  {
     int num1 = 10;
     public Test()
     {
        Console.Write(num1 + " ");
     }
```

```
    public Test(int num2)
    {
        Console.Write(num2);
    }
}
class program
{
    public static void Main()
    {
        Test obj = new Test();
        Test obj1 = new Test(10);
    }
}
```

Ans : 10 10

15) Which of the below option is FALSE related to abstract class?

[Choose most appropriate option]

a) Abstract class can contain constructors

b) Abstract class can be instantiated

c) Abstract class can have abstract and non-abstract methods

Ans: Abstract class can be instantiated

16) What will be the output for the following program?

```
using System.Collections;
using System;
public class program
{
    public static void Main(string[] args)
    {
```

```
    ArrayList fruits = new ArrayList(){"Apple", "Banana", "Orange" };
    fruits.Insert(2, "Grapes");
    fruits.Add("Bilberry");
    foreach(var item in fruits)
    {
       Console.WriteLine(item);
    }
  }
```

Ans : Apple Banana Grapes Orange Bilberry

17)

```
using System;
class Program
{
  static void Method(int[] num)
  {
    num[0]=11;
  }
  static void Main()
  {
    int[] numbers={1, 2, 3};
    Method(numbers);
    Console.WriteLine(numbers[0]);
  }
```

Options:

a) 11

b) 1

c) compilation error at line 2

d) 11 2 3

Answer: 11

18) Identify the keyword used to specify that the class cannot participate in inheritance?

a) abstract

b) sealed

c) virtual

d) override

Answer: b) Sealed

19) using System;

```
public class StaticTest
{
   static int num1=55;
   public void Display(int num)
   {
      num1 += num;
      Console.WriteLine("Value of num1 is "+ num1);
   }
}
public class Program
{
   static void Main()
   {
      StaticTest obj1= new StaticTest();
      obj1.Display(10);
      StaticTest obj2= new StaticTest();
      obj2.Display(20);
   }
}
```

Answer: Value of num1 is 65 Value of num1 is 85

20) using System;

```
class Program
{
   static void Main()
   {
      int[] first = {6, 7};
      int[] second = {1, 2, 3, 4, 5};
      second = first;
      foreach(int j in second)
      {
        Console.WriteLine(j);
      }
   }
}
```

Answer: Prints 6 7

21) Which access specifier can be accessed anywhere in the Assembly but not outside the Assembly?

a) internal

b) public

c) private

d) protected

Answer: a) internal

22) Which Feature of C# is Demonstrated by the following code ?

```
using System;
class Employee
{
   int id;
```

```
    public int Id
    {
        get { return id;}
        internal set { id = value;}
    }
}
```

Options:

a) Automatic properties

b) Read only property

c) Abstraction

d) Asymmetric property

Answer: Asymmetric property

23) Predict the output of the following code:

```
enum Numbers
One=100,
Two, Three
}
class Program
static void Main()
Console.WriteLine((int)Numbers. Three );
```

Answer : 102

24) What will be the output of following code

```
int? num= 100; num= nutt
Console.WriteLine(num.GetValueOrDefault()
```

Answer: 0(zero)

25)

```
class Program
{
```

```
static void Method(int[] num)   {
num(0] = 11
}
static void Main()
{
Int[] numbers = {1,2,3}
 Method(numbers)
Console.WriteLine(number[0]);
}}
```

Answer = 1 1

```
26) using System;
 class Demo (
 static int x;
 int y
 Demo(){
X++;
Y++;
Console.Write(x +” “ + y + " ");
}
static void Main(string[] args)
{
 Demo d1 = new Demo();
Demo d2 = new Demo();
}
```

Answer = 1 1 2 1

```
27) class Program
```

```
{
static void Main()
{
int first = (6, 7);
int[] second = {1, 2, 3, 4, 5);
second = first;
 foreach(int j in second)
{
Console.WriteLine(j);
} } }
```

Answer – 6,7

28)

```
using System:
 class Program{
void Method(int a, int b, out int c, ref int d)
{
a = a +1:
b= b+ 2;
c = a + b;
 d = c -1;
}
 static void Main()
{
int first = 10, second =20, third, fourth = 30 ;
 Program p = new Program()
p. Method(first, second, out third, ref fourth);
Console.WriteLine("{0} (1) (2) (3), first, second,third, fourth);
}}
```

Answer – 10 20 33 32

29) class Test

```
 {
 int num1 = 10;
 public Test()
{
 Console.Write(num1+" ");
}
 public Test(int num2)
{
 Console.Write(num2);
}
}
 class Program
{
public static void Main()
{
Test obj = new Test():
 Test obj1 = new Test(10):
}}
```

Answer – 10 10

30)using System;

```
Public class StaticTest
{
Static int num1=55;
Public void Display(int num)
```

```
{
num1 += num;
Console.WriteLine(“Value of num1 is “ + num1);
} }
Public class Program
{
Static void Main()
{
Static void main()
{
StaticTest obj1 = new StaticTest();
Obj1.Display(10);
StaticTest obj2 = new StaticTest();
Obj2.Display(20);
} }
```

Answer – value of num1 is 65 value of num1 is 85

31) using system:

```
class NamedParameterExample
{
static void Method(int num1, int num2, int num3=30)
{
Console.WriteLine("num1=(0), num2=(1), num3={2}", num1, num2, num3);
}
public static void Main()
{
Method(num2: 20, num1: 10, num3:100);
```

}}

Answer – num1= 10, num2=20,num3=100

32)using System;

Class DynamicDemo

{

Public static void main(string[] args)

{

Dynamic val1 =500;

Dynamic val2=”jyothi”;

Val2=val1;

Console.WriteLine(“val1={0},val2={1}”,val1.GetType(),val2.GetType());

}

}

Answer- val1=System.Int32,val2=System.Int32

33)using System;

using System.Collections;

namespace ConsoleApplicationDemo

{

```
   class Program
   {
      static void Main(string[] args)
      {
         ArrayList myList = new ArrayList();
         myList.Add(32);
         myList.Add(12);
         myList.Add(22);
         myList.Sort();
```

```
        for(int index=0; index<myList.Count; index++)
        {
            Console.Write(myList[index] + "\t");
        }
    }
  }
}
```

Answer: 12 22 32

34)

```
using System;
namespace Test_App
{
   class SampleClass
   {
      public void GreetUser()
      {
         Console.WriteLine("Hello Customer");
      }
      static void Main(string[] args)
      {
         global::Test_App.SampleClass sc= new SampleClass();
         sc.GreetUser();
         Console.ReadLine();
      }
   }
}
class SampleClass
{
```

```
   public void GreetUser()
{
Console.WriteLine("Hello Premium Customer");
}
}
```

Answer: Hello Customer

35) _____ class is used to find out object's Metadata i.e, methods, fields, properties at runtime

a) System Type

b) System Reflection

c) System Assembly

d) System String

Answer: System Reflection

36) Which of the following Attribute should be used to indicate the property must NOT be serialized while using .JSON serializer ?

a) XMLIgnore

b) IgnoreDataMember

c) IgnoreProperty

d) JsonIgnore

Answer: JsonIgnore

37)

```
using System;
public class program
{
   public static void Main(string[] args)
   {
      try
   {
      int num1=5;
```

```
        Console.WriteLine(num1);
        try
        {
            int num3= 5;
            int num4= 0;
            int num5= num3/num4;
            Console.WriteLine(num4);
        }
        catch (DivideByZeroException ex)
        {
            Console.WriteLine("Divide by zero Exception");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Inner Catch");
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("Outer Catch");
    }
    }

}
```

Answer: 5 Divide by Zero Exception

38) Which serialization serializes even private access specifier properties

a) XML

b) Binary

c) JSON

d) None of the given choices

Answer: Binary

39) Method overriding is a concept releated to ?

a) Dynamic Polymosphism

b) static polymosphism

c) abstract class

d) encapsulation

Answer: Dynamic Polymorsphism

40) Which of the following is Generic collection ?

a) List

b) ArrayList

c) Hash Table

d) String Collection

Answer: List

41) Which of the following is FALSE about Interface ?

a) The methods inside interface are abstract

b) Interface can have constructors

c) Interface can contain one or more methods

d) interface methods by default are public

Answer: Interface can have constructors

42) using System;

int index;

int value = 100;

int[] arr = new int[10];

try{

```
   Console.Write("Enter a number: ");
   index = Convert.ToInt32(Console.ReadLine());
   arr[index]= value;
}
catch (FormatException e)
{
   Console.Write("Bad Format ");
}
catch (IndexOutOfRangeException e)
{
   Console.Write("Index out of bounds ");
}
{
Console.Write("Remaining program");
}
```

Answer: Bad Format Remaining program

43)

```
using System;
class ConstructorDemo
{
   private int number= 300;
   public ConstructorDemo()
   {
      number=number+1;
      Console.Write(number+"\t");
   }
   public ConstructorDemo(int num)
   {
```

```
        number+=num;
        Console.Write(number+"\t");
    }
}
class NormalConstructorProgram
{
    static void Main(string[] args)
    {
        ConstructorDemo obj= new ConstructorDemo();
        ConstructorDemo obj1= new ConstructorDemo(200);
    }
}
```

Answer: 301 500