C-Sharp Quiz

January 1

2021

Place your answers in the accompanying ReadMe.md in the locations required.

Jan 2021 - Language Review

```
1)
2)  using System;
3)  using System.Collections.Generic;
4)  using System.Linq;
5)  using System.Text;
6)  using System.Threading.Tasks;
7)
8)  #region Additional Namespaces
9)  using OOPsData;
10) #endregion
11)
12) namespace OOPsConsoleDriver
13) {
14)     class Program
15)     {
16)         static void Main(string[] args)
17)         {
18)             List<Turns> gameTurns = new List<Turns>();
19)             Die[] dice = new Die[2];
20)             dice[0] = new Die();
21)             dice[1] = new Die(6, 3, "Green");
22)             string menuChoice = "";
23)             do
24)             {
25)                 Console.WriteLine("Game Menu: \n");
26)                 Console.WriteLine("A) Set Die side count");
27)                 Console.WriteLine("B) Roll the dice");
28)                 Console.WriteLine("C) Display all game turn results");
29)                 Console.WriteLine("X) Exit");
30)                 Console.Write("Enter menu choice: ");
31)                 menuChoice = System.Console.ReadLine();
32)
33)                 switch(menuChoice.ToUpper())
34)                 {
35)                     case "A":
36)                         {
37)                             SetDiceSides(dice);
38)                             break;
39)                         }
40)                     case "B":
41)                         {
42)                             dice[0].Roll();
43)                             dice[1].Roll();
44)                             Turns aturn = new Turns();
45)                             string winner = aturn.GetWinner(dice[0].FaceValue,
46)                                                             dice[1].FaceValue);
47)
48)                             Console.WriteLine(string.Format("Results: Player1 rolled: {0}
                                    Player2 rolled: {1} Winner: {2}",
49)                                         aturn.Player1, aturn.Player2, winner));
50)                             gameTurns.Add(aturn);
51)                             break;
52)                         }
53)                     case "C":
54)                         {
55)                             DisplayGame(gameTurns);
56)                             break;
57)                         }
58)                     case "X":
59)                         {
60)                             Console.WriteLine("Thank you for playing. Come again.");
61)                             break;
62)                         }
63)                     default:
```

```
64)                         {
65)                             Console.WriteLine("Invalid menu choice. Try again.");
66)                             break;
67)                         }
68)                 }
69)             } while (menuChoice.ToUpper() != "X");
70)             Console.ReadKey();
71)         }
72)
73)         public static void SetDiceSides(Die[] dice)
74)         {
75)             string facecount = "";
76)             int faces = 6;
77)             Console.WriteLine("Set dice face count to 6 or 12.");
78)             Console.WriteLine("An invalid entry will default to 6.");
79)             Console.Write("Enter number of sides: ");
80)             facecount = System.Console.ReadLine();
81)             if (!int.TryParse(facecount, out faces))
82)             {
83)                 Console.WriteLine("Die face " + facecount + " is invalid. Die will be set to
    6");
84)                 faces = 6;
85)             }
86)             else
87)             {
88)                 if (faces == 12)
89)                 {
90)                     Console.WriteLine("Die face count set to 12");
91)                 }
92)                 else if (faces == 6)
93)                 {
94)                     Console.WriteLine("Die face count set to 6");
95)                 }
96)                 else
97)                 {
98)                     Console.WriteLine("Die face count was invalid. Face count set to 6");
99)                     faces = 6;
100)                }
101)            }
102)            for (int x = 0; x < 2; x++)
103)            {
104)                dice[x].SetSides(faces);
105)            }
106)        }
107)
108)        public static void DisplayGame(List<Turns> gameTurns)
109)        {
110)            string msg = "";
111)            Console.WriteLine("This is the complete set of turns for this game:\n");
112)            int[] TotalWins = new int[3] { 0, 0, 0};
113)
114)            foreach(Turns aturn in gameTurns)
115)            {
116)                msg = string.Format("Results: Player1 rolled: {0} Player2 rolled: {1} Winner:
    {2}",
117)                                    aturn.Player1, aturn.Player2, aturn.Winner);
118)                Console.WriteLine(msg);
119)                if (aturn.Winner.Equals("Player1"))
120)                {
121)                    TotalWins[0]++;
122)                }
123)                else if(aturn.Winner.Equals("Player2"))
124)                {
125)                    TotalWins[1]++;
126)                }
127)                else
```

```
                {
                    TotalWins[2]++;
                }

            }
           msg = string.Format("Final Results: Player1 won: {0} Player2 won: {1} Ties: {2}",
                      TotalWins[0], TotalWins[1], TotalWins[2]);
            Console.WriteLine(msg);
            Console.WriteLine("\n");
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OOPsData
{
    public class Die
    {
        private static Random _rnd = new Random();
        private int _Sides;
        public int Sides
        {
            get { return _Sides; }
            private set{_Sides = value; }
         }

         public int FaceValue { get; private set; }

        private string _Color;
        public string Color
        {
            get{ return _Color; }
            set
            {
                if (string.IsNullOrEmpty(value))
                {
                    throw new Exception("You must supply a color");
                }
                else
                {
                    _Color = value;
                }
            }
        }
        public Die()
        {
            SetSides(6);
            Color = "White";
        }

        public Die (int sides, int facevalue, string color)
        {
            SetSides(sides);
            FaceValue = facevalue;
            Color = color;
        }

        public void SetSides(int sides)
        {
            if (!(sides == 6 || sides == 12))
            {
                throw new Exception("Invalid number of sides for the die");
```

```
196)             }
197)             else
198)             {
199)                 Sides = sides;
200)                 Roll();
201)             }
202)         }
203)
204)         public void Roll()
205)         {
206)             FaceValue = _rnd.Next(1, Sides + 1);
207)         }
208)     }
209) }
210) using System;
211) using System.Collections.Generic;
212) using System.Linq;
213) using System.Text;
214) using System.Threading.Tasks;
215)
216) namespace OOPsData
217) {
218)     public class Turns
219)     {
220)         public int Player1 { get; set; }
221)         public int Player2 { get; set; }
222)         public string Winner { get; set; }
223)
224)         public string GetWinner(int player1roll, int player2roll)
225)         {
226)             Player1 = player1roll;
227)             Player2 = player2roll;
228)             if (Player1 > Player2)
229)             {
230)                 Winner = "Player1";
231)             }
232)             else if (Player1 < Player2)
233)             {
235)                 Winner = "Player2";
236)             }
237)             else
238)             {
239)                 Winner = "Draw";
240)             }
241)             return Winner;
242)         }
243)     }
244) }
```

1) Which term best describes the boxed portion of the following code which is copied from line 2 of the attached code?

   `using System.Collections.Generic;`

   a. Class Name
   b. Namespace
   c. Method Name
   d. Object Name

2) Which term best describes the boxed portion of the following code which is copied from line 184 of the attached code?

```
public Die (int sides, int facevalue, string color)
```

a. Return Type
b. Argument
c. Parameter
d. Constructor

3) Which term best describes the boxed portion of the following code which is copied from line 232 of the attached code?

```
else if (Player1 < Player2)
```

a. Property
b. Local Variable
c. Field
d. Method Name

4) Which term best describes the boxed portion of the following code which is copied from line 21 of the attached code?

```
dice[1] = new Die(6, 3, "Green");
```

a. Data Type
b. Variable
c. Expression
d. Literal Value

5) Which term best describes the boxed portion of the following code which is copied from line 184 of the attached code?

```
public Die (int sides, int facevalue, string color)
```

a. Local Variable
b. Field
c. Property
d. Method Name
e. Constructor

6) Which term best describes the boxed portion of the following code which is copied from line 108 of the attached code?

```
public static void DisplayGame(List<Turns> gameTurns)
```

a. Local Variable
b. Field
c. Property
d. Method Name
e. Parameter

7) Which term best describes the boxed portion of the following code which is copied from line 31 of the attached code?

```
menuChoice = System.Console.ReadLine();
```

a. Class Name
b. Method Name
c. Object Name
d. Namespace

8) Which term best describes the boxed portion of the following code which is copied from line 150 of the attached code?

```
private static Random _rnd = new Random();
```

a. Method Name
b. Field
c. Property
d. Class
e. Local Variable

9) Which term best describes the boxed portion of the following code which is copied from line 19 of the attached code?

```
Die[] dice = new Die[2];
```

a. Position in Array
b. Array Variable
c. Size of Array
d. Array Element

10) Which term best describes the boxed portion of the following code which is copied from line 80 of the attached code?

```
facecount = System.Console.ReadLine();
```

a. Fully-Qualified Object Name
b. Class Name
c. Fully-Qualified Class Name
d. Object Name

11) Which term best describes the boxed portion of the following code which is copied from line 21 of the attached code?

```
dice[1] = new Die(6, 3, "Green");
```

a. Constructor Call
b. Instance Method Call
c. Global Method Call
d. Static Method Call

12) Which term best describes the boxed portion of the following code which is copied from line 18 of the attached code?

```
List<Turns> gameTurns = new List<Turns>();
```

a. Method Name
b. Class Name
c. Object Name
d. Namespace

13) Which term best describes lines 224-242 of the attached code?

a. Auto-Implemented Property
b. Constructor
c. Method
d. Fully-Implemented Property
e. Field

14) Which term best describes the boxed portion of the following code which is copied from line 83 of the attached code?

```
Console.WriteLine("Die face " + facecount + " is invalid. Die will be set to 6");
```

a. Argument
b. Return Type
c. Call to a Property Getter
d. Call to a Property Setter

15) Which term best describes the boxed portion of the following code which is copied from line 146 of the attached code?

```
namespace OOPsData
```

a. Class Name
b. Object Name
c. Namespace
d. Method Name

16) Which term best describes the boxed portion of the following code which is copied from line 112 of the attached code?

```
int[] TotalWins = new int[3] { 0, 0, 0};
```

a. Array Initializer
b. Value Body
c. Placeholder
d. Expression

17) Which term best describes the boxed portion of the following code which is copied from line 104 of the attached code?

```
dice[x].SetSides(faces);
```

a. Global Method Call
b. Constructor Call
c. Static Method Call
d. Instance Method Call

18) Which term best describes the boxed portion of the following code which is copied from line 224 of the attached code?

```
public string GetWinner(int player1roll, int player2roll)
```

a. Argument
b. Return Type
c. Parameter
d. Constructor

19) Which term best describes line 160 of the attached code?

a. Fully-Implemented Property
b. Field
c. Auto-Implemented Property
d. Constructor
e. Method

20) Which term best describes lines 161-175 of the attached code?

a. Field
b. Constructor
c. Method
d. Auto-Implemented Property
e. Fully-Implemented Property

21) Which term best describes the boxed portion of the following code which is copied from line 83 of the attached code?

```
Console.WriteLine("Die face " + facecount + " is invalid. Die will be set to 6");
```

a. Literal Value
b. Concatenation
c. Parameter
d. Data Type

22) Which term best describes the boxed portion of the following code which is copied from line 83 of the attached code?

```
Console.WriteLine("Die face " + facecount + " is invalid. Die will be set to 6");
```

a. Assignment Statement
b. Literal Value
c. Expression
d. Method Call

23) Which term best describes the boxed portion of the following code which is copied from line 224 of the attached code?

```
public string GetWinner(int player1roll, int player2roll)
```

a. Data Type
b. Variable
c. Literal Value
d. Expression

24) Which term best describes the boxed portion of the following code which is copied from line 224 of the attached code?

```
public string GetWinner(int player1roll, int player2roll)
```

a. Literal Value
b. Variable
c. Data Type
d. Expression

25) Which term best describes the boxed portion of the following code which is copied from line 44 of the attached code?

```
Turns aturn = new Turns();
```

a. Object Name
b. Class Name
c. Method Name
d. Namespace

26) Which term best describes the boxed portion of the following code which is copied from line 206 of the attached code?

```
FaceValue = _rnd.Next(1, Sides + 1);
```

a. Data Type
b. Variable
c. Literal Value
d. Expression

27) Which term best describes the boxed portion of the following code which is copied from line 21 of the attached code?

```
dice[1] = new Die(6, 3, "Green");
```

a. Size of Array
b. Position in Array
c. Array Variable
d. Array Element

28) Which term best describes the boxed portion of the following code which is copied from line 21 of the attached code?

```
dice[1] = new Die(6, 3, "Green");
```

a. Literal Value
b. Data Type
c. Variable
d. Expression

29) Which term best describes the boxed portion of the following code which is copied from line 21 of the attached code?

```
dice[1] = new Die(6, 3, "Green");
```

a. Array Variable
b. Array Element
c. Size of Array
d. Position in Array

30) Which term best describes the boxed portion of the following code which is copied from line 21 of the attached code?

```
dice[1] = new Die(6, 3, "Green");
```

a. Array Variable
b. Array Element
c. Size of Array
d. Position in Array

31) Which term best describes line 158 of the attached code?

a. Constructor
b. Method
c. Fully-Implemented Property
d. Field
e. Auto-Implemented Property

32) Which term best describes the boxed portion of the following code which is copied from line 135 of the attached code?

```
Console.WriteLine(msg);
```

a. Static Method Call
b. Instance Method Call
c. Global Method Call
d. Constructor Call

33) Which term best describes lines 178-182 of the attached code?

a. Method
b. Field
c. Constructor
d. Auto-Implemented Property
e. Fully-Implemented Property