Student Name: Zach Carie

IT Foundation using C# EXAM 1 Instructor: Vallejo

Note: The test is worth **100 points**. **Show all your work** for each problem. No partial credit will be given if no work is shown for each answer. Read the entire description to each question before answering the question. **Good Luck!**

True / False (2 points each)

Circle One

1. A block (**{ }**)can contain more than one statement. [✓ TRUE] / FALSE
2. Every program must have a function called Main. [✓ TRUE] / FALSE
3. The type **int** is signed. [✓ TRUE] / FALSE
4. Multi-line comments are started by **//**. TRUE / [✓ FALSE]
5. Variables are used only for storing constants. TRUE / [✓ FALSE]
6. All statements are terminated by a comma. TRUE / [✓ FALSE]
7. A variable name may begin with an underscore (_). [✓ TRUE] / FALSE
8. **\n** is used by **WriteLine** to go to the next new line. [✓ TRUE] / FALSE
9. Upper- and lower-case letters are significant for names. [✓ TRUE] / FALSE
10. The type **char** is Unicode (2 bytes). [✓ TRUE] / FALSE

Multiple Choice (3 points each)

11. Which feature will execute a block of code at least once:
    - A. **while**
    - B. **for**
    - C. **do-while**
    - [✓ D. **foreach**]

12. An **int** variable occupies:
    - A. One byte
    - B. 9 bits (1 for parity)
    - [✓ C. Four bytes]
    - D. 7 bits (for unsigned)

13. What function is used to read in a string:
    - A. Console.WriteLine
    - B. Console.Read
    - C. Console.Write
    - [✓ D. Console.ReadLine]

14. Which is an INVALID statement:

A. `x = x / -1;`
[✓ B. `y = y + 2`] no semicolon
C. `z = z + z;`
D. `t += t;`

15. What does X == Y mean?

A. X is assigned to Y
B. Y is assigned to X
[✓ C. X is compared to Y]
D. None of the above

16. Which of the following is NOT a logical operator:

[✓ A. ;]
B. ||
C. &&
D. !

17. Which is an invalid type of Field:

A. readonly
[✓ B. get]
C. const
D. None of the above

18. **break** is used to:

A. Exit a program
B. Exit stage right
C. Exit a function
[✓ D. Exit a loop] or switch statement

19. C# ignores:

[✓ A. Whitespace]
B. Braces
C. Commas
D. Semicolons

20. What is the significance of **while (true)**

A. It is an invalid expression
B. It is an infinite trip
[✓ C. It is an infinite statement]
D. It will never stop

21. What is the result of the following statement for x = 4? (5 points)

```
result = X + X++;
result = --X + X;
```

= 8 (4+4++) the second "X" variable doesn't change until after the calculation.

= 6 (3+3) x will change to "3" before the calc

22. What are the basic arithmetic operations? Show the operational signs. (5 points)

+, -, %, *, /

23. What are the basic conditional operations (less than, equality, etc.)? <, >, >=, <=, ==, != What are the basic logical operations? (5 points)

||, &&, !

24. Check if the following *if* expressions below result in TRUE or FALSE? (5 points)

A.

```
usCnt = 10; usSum = 10;
if (usSum++ == usCnt)
   {
   etc...
   }
```

[✓ TRUE] FALSE

B.

```
usCnt = 10; usSum = 10;
if (usSum == ++usCnt)
{
etc...
}
```

TRUE [✓ FALSE]

25. What is the value of usSum after the following code segment? (5 points)

```
usSum = 10; usCnt = 2;
switch (usCnt)
   {
   case 3:
      {
      usSum = usSum + 2;
      break;
      }
   default:
      {
      break;
      }
   case 2:
      {
      usSum = usSum * 3;
      goto case 3;
      }
   }
```

usSum = 32

It will start at case 2: which multiplies 10 * 3 = 30. Then goes to case 3 which adds 2 = (30+2=32)

26. *Circle* the COMPILER/SYNTAX errors in this program (5 points)

```
static void Main(}  paraenthesis
{
int    iSum,;
int    iCnt=Sum, iValue; iTotal;
char   chChar = "a";  single quotes

iSum = chChar
;
while (iSum = 100);
   {
   iSum = iSum + 1;
   }
)
```

27. Given the following program what will the last value of usCnt be when the program completes execution? Is there anything unusual about this program? What does it show and what do you conclude from this? (10 points)

```
using System;
class Test
{
   static void Main( )
   {
   uint  usCnt;
   uint  usSum = 0;

   for (usCnt = 10; usCnt >= 0; usCnt--)
      {
      Console.WriteLine("{0}",usCnt);
      usSum = usSum + usCnt;
      }
   }
}
```

I do not believe this program will ever finish. Using "uint" as the integer type means it will never go negative. This also means the for loop will always be true. So usCnt will count down by 1 until it hits zero then jump to its largest value. usSum will add those values to itself until it hits it's max value ~~then count up again~~ then start again from 0 and continue to loop around.

28. What will the following program display? (10 points)

```
using System;
class Test
{
    static void Main( )
    {
    int    iX;
    int    iY;

    iX = 15321;
    while (iX != 0)
        {
        iY = iX % 10;
        Console.Write(iY);
        iX /= 10;
        }
    Console.WriteLine();
    }
}
```

15321 / 10 = 1532.1 iX = 1532 iY = 1
1532 / 10 = 153.2 iX = 153 iY = 2
153 / 10 > 15.3 iX = 15 iY = 3
15 / 10 > 1.5 iX = 1 iY = 5
1 / 10 = .1 iX = 0 iY = 1
0 / 10 = 0 iX = 0 iY = 0

$\frac{15321}{10}$ = 1532.1 So iY = 1 (the remainder)

since iX is a whole number 1532.1 = 1532

Display will be 12351 % divides and shows the remainder.

29. We have a stack object (10 points):

- What is while ( ) statement do? Executes the function if the stack is not empty
- What is IsEmpty? Checks to see if the stack has any values.
- What is Pop()? removes a value from the stack
- What is {0}? memory place holder
- What is the code below going to do? display " Popping (whatever the value from the stack is) "

```
while (!stack.IsEmpty)

    {

        Console.WriteLine("Popping {0}", stack.Pop());

    }
```

30. What is the difference between a "Class" and a "Struct" in C#? (10 points)

a Class is a reference and a struct is a value.
So a class is like a pointer to where a value is stored in memory, where a struct stores the value directly.