

Student Name:

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)

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

Circle One

- | | | |
|--|---|--------------------------------|
| <input checked="" type="checkbox"/> TRUE | / | <input type="checkbox"/> FALSE |
| <input checked="" type="checkbox"/> TRUE | / | <input type="checkbox"/> FALSE |
| <input checked="" type="checkbox"/> TRUE | / | <input type="checkbox"/> FALSE |
| <input checked="" type="checkbox"/> TRUE | / | <input type="checkbox"/> FALSE |
| <input checked="" type="checkbox"/> TRUE | / | <input type="checkbox"/> FALSE |
| <input checked="" type="checkbox"/> TRUE | / | <input type="checkbox"/> FALSE |
| <input checked="" type="checkbox"/> TRUE | / | <input type="checkbox"/> FALSE |
| <input checked="" type="checkbox"/> TRUE | / | <input type="checkbox"/> FALSE |
| <input checked="" type="checkbox"/> TRUE | / | <input type="checkbox"/> FALSE |
| <input checked="" type="checkbox"/> TRUE | / | <input type="checkbox"/> 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. $y = y + 2$
 - 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
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 = 4 + 4 = 8, x = 4 + 1 = 5
result = --X + X;    x = 5 - 1 = 4, result = 4 + 4 = 8
```

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;    32
        break;
    }
    default:
    {
        break;
    }
    case 2:
    {
        usSum = usSum * 3;    30
        goto case 3;
    }
}
```

`usSum` = 32

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

```
static void Main( )
{
    int iSum;
    int iCnt; Sum, iValue; iTotat;
    char chChar = 'a';

    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; 10, 19, 27, 34, 40, 45, 49, 52, 54, 55, 55
        }
    }
}
```

0	10
10	9
19	8
...	...
55	0

usCnt = 0

edit: Apparently since it is an unsigned integer, after 0 it goes back to 4,294,967,295 so this is a never ending loop counting down. For this to stop at 0, usCnt would have to be a regular int or change the condition to usCnt > 0.

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; 1,2,3,5,1
            Console.Write(iY); 12351
            iX /= 10; 1532,153,15,1,0
        }
        Console.WriteLine();
    }
}
```

Display will be 12351

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

- What is while () statement do? run code WHILE logic in () is true
- What is IsEmpty? returns true when stack is empty(length = 0)
- What is Pop()? takes out the top object in stack and returns it
- What is {0}? a place holder in string where object is inserted
- What is the code below going to do?

```
while (!stack.IsEmpty)
{
    Console.WriteLine("Popping {0}", stack.Pop());
}
```

e.g. stack: x, y, z, ...
write on console: "Popping x"
"Popping y"
"Popping z"
... until stack is empty

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

Classes group related methods and data together, while structures only group related data together.