Student Name: Sandeep Kanapala

|  |  |  |
| --- | --- | --- |
| 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

      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 = --X + X;

Answer

using System;

//++ and -- operator

class MainClass

{

static void Main()

{

int x;

int result;

x = 4;

result = x + x++;

Console.WriteLine("the value of result is {0}", result); //result is 8

x = 4;

result = --x + x;

Console.WriteLine("the value of result is {0}", result); //result is 6

Console.ReadLine();

}

}

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

Answer

using System;

class Program

{

static void Main(string[] args)

{

int a = 10;

int b = 5;

int resultadd, resultsub, resultmul, resultdiv;

resultadd = a + b;

Console.WriteLine(resultadd);

resultsub = a - b;

Console.WriteLine(resultsub);

resultmul = a \* b;

Console.WriteLine(resultmul);

resultdiv = a / b;

Console.WriteLine(resultdiv);

Console.ReadLine();

}

}

23.  What are the basic conditional operations (less than, equality, etc.)?

What are the basic logical operations? (5 points)

Answers: Conditional operations are >, <, >=, <+, ==. Logical operations are &, ||, ^.

using System;

class operators

{

static void Main()

{

int x = 5;

int y = 2;

if (x > 1 && y > 1)

{

Console.WriteLine("x is greater than 1 AND y is greater than 1");

}

if (x > 5 || y < 4)

{

Console.WriteLine("x is greater than 5 OR y is less than 4");

}

if ((x > 2 && y < 3) || x == 2)

{

Console.Write(" (x is greater than 2 AND y is less than 3)");

Console.WriteLine(" OR (x is equal to 2)");

}

}

}

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

A.                                                         B.

   usCnt = 10; usSum = 10;        usCnt = 10; usSum = 10;

   if (usSum++ == usCnt)          if (usSum == ++usCnt)

      {                           {

      etc...                      etc...

      }                           }

      TRUE        FALSE                              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\_\_\_\_

using System;

namespace examprob25

{

class Program

{

static void Main(string[] args)

{

int usSum = 10; int usCnt = 2;

switch (usCnt)

{

case 3:

{

usSum = usSum + 2;

Console.WriteLine(usSum);

Console.ReadLine();

break;

}

default:

{

break;

}

case 2:

{

usSum = usSum \* 3;

goto case 3;

}

}

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

Answer: Highlighted errors with green

   static void Main( }

   {

   int   iSum,;

int   iCnt=Sum, iValue; iTotal;

   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)

Answer: Since the usCnt is continuously decrementing and checking for greater than equal to condition which result in true condition every time the condition is checked, it is resulting in infinite loop. An result is displayed on every execution and is never ending. We should be careful while declaring conditions in loops.

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;

      }

}

}

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();

}

}

Display will be \_\_\_12351\_\_\_\_\_\_\_

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

* What is while ( ) statement do?
* What is IsEmpty?
* What is Pop()?
* What is {0}?
* What is the code below going to do?

      while (!stack.IsEmpty)

{

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

}

 Answer:

* The while statement executes a statement continuously until it is false. In this case, while stack is not empty, execute the loop.
* IsEmpty is a method defined to check if the stack is empty or not. A method that has a piece of code to check the stack status (empty or not).
* Pop is a method written in the code and will be called to execute a piece of code defined in it. In the this case (class excursive done on session 8) it will decrement the stack value by 1 and then returns the value.
* {0} is a placeholder to display the popped stack number onto the console.
* while stack is not empty, execute the loop. Write the stack numbers pushed into the array onto the console.

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

Answer: Class are the basis for object-oriented languages that allows to group related methods and data into a single unit.

A **struct** is used to group related data into a single entity as well just as class with constructors, constants, methods, etc.

The difference between a **struct** and a **class** is that you don't need to call **new** unless there is a constructor.

And unlike structs, classes support inheritance. We cannot inherit from a **struct** nor can a **struct** inherit from another type.