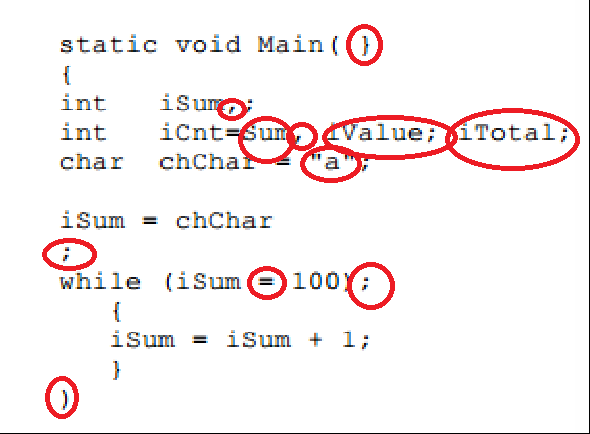
1. True
2. True
3. False
4. False
5. False
6. False
7. False
8. True
9. True
10. True
11. B. for
12. C. Four bytes
13. D. Console.ReadLine
14. D. t += t;
15. C. X is compared to Y
16. A. ;
17. B. get
18. D. Exit a loop
19. A. Whitespace
20. C. It is an infinite statement
    1. “result = X + X++;” will add X + X and store that as result, then increase the value of X by 1, in this case “result” = 8 and “X” = 5 after the first line has executed
    2. “result = --X + X;” will decrease the value of x by one, then add X + X, in this case “X” would be reduced to 4, then “result” = 4 + 4 = 8
21. Arithmetic operators
    1. + is the addition operator
    2. – is the subtraction operator
    3. \* is the multiplication operator
    4. / is the division operator
22. Logical operators
    1. < less than
    2. <= less than or equal to
    3. > greater than
    4. >= greater than or equal to
    5. == equal to
    6. != not equal to
23. Check the *if* expressions are TRUE or FALSE
    1. TRUE
       1. The condition of usSum == usCnt will be checked before the usSum variable will be incremented
    2. FALSE
       1. The usCnt variable will be incremented (to 11), then the condition will be checked resulting in a false statement
24. The code will move to “case 2”, which will set usSum to 30, then the code will move to “case 3” where usSum will be set to 32. The final result of usSum will be usSum = 32.
25. 
26. The program will never complete its execution. “usCnt” is set as an unsigned integer and will therefor always be >= 0. When the variable “usCnt” is equal to 0 the loop will decrement the value to approximately 4.2 billion and continue decrementing until it reaches 0 again and do the same thing. The variable should be assigned to a type of int. This would allow the value of “usCnt” to go below 0 creating a false condition where the loop would be exited.
27. The display will show “12351” The program will assign a value of 15321 to iX, then move into a loop where the condition iX != 0 will be checked before each iteration. Each iteration of the loop will set iY to the remainder when dividing the current iX by 10, Then write that value to the console, finally the loop will divide iX by 10 then set that value as the new iX. The loop will be executed once iX = 0 after 5 iterations.
28. Stack of objects.
    1. The while() statement will begin a loop that will execute while the condition inside it true.
    2. IsEmpty is checking if the stack has any values in it. While there are values in the stack this will result in TRUE, when there are no values in the stack this will return as FALSE.
    3. Pop() will return the value in the highest position in the stack then move our pointer to the nest position (down in the stack counter).
    4. {0} is a reference in the Console.Writeline string which will be replaced in the console statement with the value popped from the stack.
    5. The code will write each value which is in the stack to the console in the opposite order they were input (LIFO)
29. A struct is a more simplified version of a class. Classes supports inheritance and polymorphism, while structs cannot. Another major difference is the memory used between the two. A Struct is stored on the stack, resulting in faster access. Classes are stored on the heap, meaning memory is assigned at runtime and access can be slower.