Здравейте,   
  
искам да споделя няколкото онлайн C# теста, които открих. Първият е разделен по категории:  
  
<http://www.indiabix.com/c-sharp-programming/questions-and-answers/>  
  
другите два са съответно  
  
<http://www.wiziq.com/online-tests/17184-dotnet-with-c-test-quiz>  
  
и <http://www.wiziq.com/online-tests/26849-c-basic-test>  
  
На мен лично - първият много ми допадна, защото въпросите наистина са за начинаещи.   
  
Надявам се да са ви полезни и на вас . Успех на всички!

http://forums.academy.telerik.com/27450/%D1%82%D0%B5%D1%81%D1%82%D0%BE%D0%B2%D0%B5-%D0%B7%D0%B0-%D1%83%D0%BF%D1%80%D0%B0%D0%B6%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F-c%23-%D0%B8-html

|  |
| --- |
| **1. C# (C-Sharp) was developed by** |
| http://www.academictutorials.com/images/cross.gif  Sun Microsystems |
| http://www.academictutorials.com/images/tick.jpg  Microsoft |
| http://www.academictutorials.com/images/cross.gif  Intel |
| http://www.academictutorials.com/images/cross.gif  IBM |
| **2. What is C# (C-Sharp) ?** |
| http://www.academictutorials.com/images/cross.gif  An Operating System |
| http://www.academictutorials.com/images/cross.gif  An Programming Tool |
| http://www.academictutorials.com/images/tick.jpg  An Programming Language |
| http://www.academictutorials.com/images/cross.gif  None of these |
| **3. C# (C-Sharp) belong to which framework ?** |
| http://www.academictutorials.com/images/cross.gif  Java |
| http://www.academictutorials.com/images/tick.jpg  .NET |
| http://www.academictutorials.com/images/cross.gif  It does't belong to any framework |
| http://www.academictutorials.com/images/cross.gif  None of these |
| **4. What is CLR ?** |
| http://www.academictutorials.com/images/cross.gif  Conventional Language Runtime |
| http://www.academictutorials.com/images/cross.gif  Clarified Language Runtime |
| http://www.academictutorials.com/images/tick.jpg  Common Language Runtime |
| http://www.academictutorials.com/images/cross.gif  Colloquial Language Runtime |
| **5. Object Oriented Programming (OOP) is a style of programming in which your code is broken up into units, known as** |
| http://www.academictutorials.com/images/cross.gif  objects |
| http://www.academictutorials.com/images/cross.gif  classes |
| http://www.academictutorials.com/images/cross.gif  objects or classes |
| http://www.academictutorials.com/images/tick.jpg  objects and classes |
| **6. There are, however, some advanced features of C++, such as \_\_ that are not supported in .NET, and are therefore, not available in managed code.** |
| http://www.academictutorials.com/images/tick.jpg  templates |
| http://www.academictutorials.com/images/cross.gif  Runtime classes |
| http://www.academictutorials.com/images/cross.gif  base classes |
| http://www.academictutorials.com/images/cross.gif  multiple-document-interfaces |
| **7. Using C# you can, for example, write a \_\_\_ web page** |
| http://www.academictutorials.com/images/cross.gif  dynamic |
| http://www.academictutorials.com/images/tick.jpg  static |
| http://www.academictutorials.com/images/cross.gif  standard |
| http://www.academictutorials.com/images/cross.gif  classical |
| **8. Using C# you can, for example, write \_\_\_ of a distributed application** |
| http://www.academictutorials.com/images/cross.gif  a core |
| http://www.academictutorials.com/images/tick.jpg  a component |
| http://www.academictutorials.com/images/cross.gif  an array |
| http://www.academictutorials.com/images/cross.gif  a structure |
| **9. This is because they all have the same familiar Windows \_\_\_ at their core.** |
| http://www.academictutorials.com/images/cross.gif  GUID |
| http://www.academictutorials.com/images/tick.jpg  IDE |
| http://www.academictutorials.com/images/cross.gif  DLL |
| http://www.academictutorials.com/images/cross.gif  API |
| **10. What is OLE ?** |
| http://www.academictutorials.com/images/tick.jpg  Object Linking and Embedding |
| http://www.academictutorials.com/images/cross.gif  Oriented Linking and Embedding |
| http://www.academictutorials.com/images/cross.gif  Object Location and Embedding |
| http://www.academictutorials.com/images/cross.gif  Objective Location and Embedding |
| **11. What actually manages your code ?** |
| http://www.academictutorials.com/images/cross.gif  JIT |
| http://www.academictutorials.com/images/tick.jpg  CTS |
| http://www.academictutorials.com/images/cross.gif  CLR |
| http://www.academictutorials.com/images/cross.gif  CLS |
| **12. When the . NET runtime loads and runs code, this is the language that it expects to find the code in.** |
| http://www.academictutorials.com/images/cross.gif  Intermediate Language |
| http://www.academictutorials.com/images/cross.gif  Common Type System |
| http://www.academictutorials.com/images/tick.jpg  Just-in-Time Compilation |
| http://www.academictutorials.com/images/cross.gif  Globally Unique IDentifiers |
| **13. When you compile managed code, the \_\_\_\_ actually emits IL.** |
| http://www.academictutorials.com/images/cross.gif  debugger |
| http://www.academictutorials.com/images/cross.gif  collector |
| http://www.academictutorials.com/images/tick.jpg  translator |
| http://www.academictutorials.com/images/cross.gif  compiler |
| **14. \_\_\_\_ this is a minimum set of standards that guarantees that code can be accessed from any language.** |
| http://www.academictutorials.com/images/tick.jpg  Common Type System |
| http://www.academictutorials.com/images/cross.gif  Common Language Specification |
| http://www.academictutorials.com/images/cross.gif  Just-in-Time Compilation |
| http://www.academictutorials.com/images/cross.gif  The .NET Runtime |
| **15. Common Language Specification: This is a minimum set of standards that guarantees that code can be accessed from any** |
| http://www.academictutorials.com/images/cross.gif  compiler |
| http://www.academictutorials.com/images/cross.gif  stage |
| http://www.academictutorials.com/images/tick.jpg  application |
| http://www.academictutorials.com/images/cross.gif  language |
| **16. Just-in-Time (JIT) Compilation: This is the term for the process of performing the \_\_\_ stage of compilation from IL into native machine code.** |
| http://www.academictutorials.com/images/tick.jpg  intermediate |
| http://www.academictutorials.com/images/cross.gif  beginning |
| http://www.academictutorials.com/images/cross.gif  final |
| http://www.academictutorials.com/images/cross.gif  whole |
| **17. The function call generated by a non-OOP compiler causes what is called \_\_\_ binding** |
| http://www.academictutorials.com/images/cross.gif  weak |
| http://www.academictutorials.com/images/tick.jpg  strong |
| http://www.academictutorials.com/images/cross.gif  early |
| http://www.academictutorials.com/images/cross.gif  late |
| **18. You create a string reference:** |
| http://www.academictutorials.com/images/cross.gif  string {s}; |
| http://www.academictutorials.com/images/cross.gif  string [s]; |
| http://www.academictutorials.com/images/cross.gif  string (s); |
| http://www.academictutorials.com/images/tick.jpg  string s; |
| **19. When you create a reference, you want to connect it with a new object. You do so, in general, with the** |
| http://www.academictutorials.com/images/cross.gif  single quote |
| http://www.academictutorials.com/images/cross.gif  double quote |
| http://www.academictutorials.com/images/tick.jpg  round brackets |
| http://www.academictutorials.com/images/cross.gif  new keyword |
| **20. There are six different places to store data. This is a general-purpose pool of memory (also in the RAM area) where all C# objects live.** |
| http://www.academictutorials.com/images/cross.gif  Registers |
| http://www.academictutorials.com/images/cross.gif  The heap |
| http://www.academictutorials.com/images/cross.gif  Static storage |
| http://www.academictutorials.com/images/tick.jpg  The stack |