

1.	Como é feita a entrada e saida de dados no	C#?	
A	leia() e escreva()	В	input() e print()
V	Console.ReadLine(); e Console.WriteLine();		
2.	O que significa uma linguagem ser fortemen	nte ti	pada?
A	Não possue tipos para definir suas variaveis	/	Variáveis e constantes têm um tipo, assim como cada expressão que é avaliada como um valor
С	Somente algumas variaveis possuem atributos tipados		
3.	C# possui 4 tipos de variaveis, sendo elas: I literais e tipos genéricos.	Γipos	internos, tipos personalizados, tipos
N	Verdadeiro	В	Falso
4.	A definição: Representam inteiros, valores caracteres de texto, valores decimais e outr variavel?	_	, –
Y	Tipos Internos	В	Tipos Personalizados
С	Tipos Literais	D	Tipos genéricos
5.	A definição: Utilizado para criar seus propri de variavel?	ios ti	pos personalizados. É aplicada a qual tipo
A	Tipos Internos	V	Tipos Personalizados
С	Tipos Literais	D	Tipos genéricos

6.	A definição: Você pode especificar como um literal numérico deve ser digitado anexando uma letra ao final do número. É aplicada a qual tipo de variavel?			
A	Tipos Internos	В	Tipos Personalizados	
Ø	Tipos Literais	D	Tipos genéricos	
7.	A definição: Um tipo pode ser declarado como um espaço reservado para o tipo o variavel?		· · · · · · · · · · · · · · · · · · ·	
A	Tipos Internos	В	Tipos Personalizados	
С	Tipos Literais	V	Tipos genéricos	
8.	É possivel declarar variáveis de diferentes formas no C#, quais das alternativas abaixo esta declarando corretamente a variavel nome?			
A	string nome;	В	string nome = "Gustavo Boaz";	
С	string nome = Console.ReadLine();	D	string nome = "Gustavo" + " " + "Boaz";	
9.	Os Operadores (+, -, *, /, %), são operadores:			
A	Lógico	В	Relação	
С	Atribuição	V	Aritiméticos	
10.	Os Operadores (=, +=, -=, *=, /=, %=), são operadores:			
A	Lógico	В	Relação	
	Atribuição	D	Aritiméticos	
11.	Os Operadores (==, !=, <, <=, >, >=), são operadores:			
A	Lógico	Y	Relação	
С	Atribuição	D	Aritiméticos	
12.	Os Operadores (&&,), são operadores	:		
X	Lógico	В	Relação	
С	Atribuição	D	Aritiméticos	

13. Quais as finalidades de escrever uma boa nomenclatura ao desenvolver um código?



Criam uma aparência consistente para o código, para que os leitores possam se concentrar no conteúdo e não no layout.



Permitem que os leitores entendam o código com mais rapidez, fazendo suposições com base na experiência anterior.

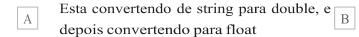


Facilitam a cópia, a alteração e a manutenção do código.



Demonstram as práticas recomendadas do C#.

14. O que esta acontecendo no trecho de código: float preco = (float) Convert.ToDouble(Console.ReadLine());



Esta convertendo de float para double, e depois fazendo um cast para string



Esta convertendo de string para double, e depois fazendo um cast para float