

Resposta 01

A, C.

Um nome de identificador deve começar com uma letra, cifrão (\$) ou sublinhado (_). Números são permitidos apenas para caracteres subsequentes. Portanto, a opção C não é um nome de variável válida. Além disso, um identificador pode não ser um único sublinhado tornando a opção A inválida para nome de variável.

Resposta 02

A.

Em uma expressão ternária, apenas uma das duas expressões mais à direita é avaliada. Desde `gorjeta > 6` é falso, `tip--` é avaliado e `tip++` é ignorado. O resultado é que a `gorjeta` mudou de 2 para 1, tornando a opção A a resposta correta. O valor do total é 7, pois o operador de pós-decremento foi usado na ponta, embora você não precise saber disso para resolver a questão.

Resposta 03

A. O f em 4.0f significa que o tipo é float, tornando a opção A correta. Tipo de variável local a inferência escolhe uma correspondência exata em vez de usar o autoboxing para escolher Float.

Resposta 04

D.

Pergunta capciosa. Não há nenhum método reverse na classe String. Há um na classe StringBuilder. Portanto, o código não compila e a opção D está correta.

Resposta 05

D.

Um `StringBuilder` é mutável, portanto, o comprimento é 2 após a conclusão da linha 6. Os métodos `StringBuilder` retornam uma referência ao mesmo objeto, para que você possa encadear o método chamadas. Portanto, `line` e `anotherLine` referem-se ao mesmo objeto. Isso significa que a linha 7 imprime verdadeiro. Então, na linha 9, ambas as referências apontam para o mesmo objeto de comprimento 2 e a opção D está correto.

Resposta 06

E.

O diagrama representa todos os casos em que maçãs ou laranjas é verdadeiro, mas bananas é false, tornando a opção E correta. A opção A está próxima, mas está correta apenas se a sobreposição superior porção de maçãs e laranjas foi preenchida. Para se divertir, você deve tentar desenhar os diagramas isso representaria as outras respostas.

Resposta 07

A.

Uma String é imutável, então um objeto diferente é retornado na linha 6. O objeto `otherLine` aponta para é de comprimento 2 após a conclusão da linha 6. No entanto, a linha original a referência ainda aponta para um objeto de comprimento 1. Portanto, a opção A está correta.

Resposta 08

B, E.

As opções A e D estão incorretas porque byte e short não armazenam valores com pontos decimais. A opção C é tentadora. No entanto, 3.14 é automaticamente um duplo. Isso requer lançando para flutuar ou escrevendo 3.14f para ser atribuído a um flutuador. Portanto, a alternativa B está correta.

Além disso, a opção E está correta porque a inferência de tipo de variável local é capaz de determinar o tipo é duplo.

Resposta 09

A, B.

A opção A está correta e lista os operadores no mesmo nível ou em um nível crescente de operador precedência. Na opção B, os três operadores realmente têm a mesma precedência de operador, então isso está correto. A opção C está incorreta, pois a divisão (/) tem precedência menor do que o operador de decremento (--). A opção D está incorreta porque o operador de complemento lógico (!) uma ordem de precedência mais alta do que os outros dois operadores. A opção E lista os operadores em a ordem correta, mas eles não se encaixam entre not equals (!=) e o operador de incremento (++) conforme listado na pergunta. Em particular, operador de adição composta (+ =) e curto-circuito o operador lógico (&&) tem uma precedência menor do que o operador diferente (!=). Finalmente, a opção F está incorreta porque o operador relacional (<) não se encaixa entre a multiplicação operador (*) e o operador de divisão (/) em ordem de precedência.

Resposta 10

D.

A linha 18 compila porque nenhum tipo é especificado para os parâmetros lambda. Linhas 19 e 22 compilam porque os parâmetros lambda usam um tipo ou var consistentemente. Estes são os três linhas que compilam, tornando a opção D correta. As linhas 20 e 21 não compilam porque var deve ser usado para todos os parâmetros em um lambda se for usado para qualquer