Program Quiz #1

Declare a variable hoursWorked and assign it to a real number value entered from the keyboard.

The correct answer is: double hoursWorked = double.Parse(Console.ReadLine());

Declare a variable payRate and assign it to a monetary value entered from the keyboard.

The correct answer is: decimal payRate = decimal.Parse(Console.ReadLine());

Declare a variable guess that can store a whole number.

The correct answer is: int guess;

Assume that you have 3 variables that can store monetary values, price, amountTendered and change.  Write an assignment statement that calculates the change due to a customer from the price and the amountTendered.  Store the result in the variable change.

The correct answer is: change = amountTendered - price;

Assume that you have 3 variables that can store real numbers, miles, gallons and milesPerGallon.  Write an assignment statement that calculates miles per gallon from miles and gallons and store the result in the variable milesPerGallon.

The correct answer is: milesPerGallon = miles / gallons;

Declare a variable payRate that can store money.

The correct answer is: decimal payRate;

Declare a variable guess and assign it to an integer value entered from the keyboard.

The correct answer is: int guess = int.Parse(Console.ReadLine());

Assume that you have 2 integer variables, miles and hours, and one double variable mph.  Write an assignment statement that calculates the miles per hour and that does NOT truncate the results.  Store the result in the variable mph.

The correct answer is: mph = (double) miles / hours;

Declare a variable hoursWorked that can store a real number.

The correct answer is: double hoursWorked;

QUIZ 3

Assume that you have already declared and created a random number generator called gen (Random gen = new Random();).  Write a statement that declares an integer variable computerChoice and assigns it to a random integer between 1 and 3.

The correct answer is: int computerChoice = gen.Next(1, 4);

Write the "opening line" of an if statement that checks to see if the value in the variable number is even.  Don't include the opening curly brace!

The correct answer is: if (number % 2 == 0)

Write the "opening line" of an if statement that checks to see if the value in the variable x is not equal to 0.  Don't include the opening curly brace!

The correct answer is: if (x != 0)

The keyword that MUST be at the end of each case that does not "fall through" in a switch statement.The correct answer is: break

Write the "opening line" of an if statement that checks to see if the value in the variable number is greater than 100.  Don't include the opening curly brace!

The correct answer is: if (number > 100)

The keyword that begins the part of an if statement that executes when none of the other conditions is true.The correct answer is: else

Assume that you have already declared and created a random number generator called gen.  (Random gen = new Random();) Write a statement that declares an integer variable secretNumber and assigns it to a random integer between 1 and 10.

The correct answer is: int secretNumber = gen.Next(1, 11);

Write the "opening line" of an if statement that checks to see if the value in the string variable choice is the word Rock.  Don't include the opening curly brace!

The correct answer is: if (choice == "Rock")

Write the "opening line" of a switch statement that compares the value in the variable x to a finite set of values.  Don't include the opening curly brace!

The correct answer is: switch (x)

Write the "opening line" of an if statement that checks to see if the value in the variable number is divisible by 5.  Don't include the opening curly brace!

The correct answer is: if (number % 5 == 0)

Write the case statement that executes when none of the other cases are true in a switch statement. The correct answer is: default:

Write the "opening line" of an if statement that checks to see if the value in the variable number is odd.  Don't include the opening curly brace!

The correct answer is: if (number % 2 != 0)

Write the "opening line" of an if statement that checks to see if the value in the variable number is between 1 and 10 inclusive.  Don't include the opening curly brace!

The correct answer is: if (number >= 1 && number <= 10)

Write the "opening line" of a switch statement that compares the value in the variable choice to a finite set of values.  Don't include the opening curly brace!

The correct answer is: switch (choice)

Write the "opening line" of an if statement that checks to see if the value in string the variable choice is Yes or yes .  Don't include the opening curly brace!

The correct answer is: if (choice == "Yes" || choice == "yes")

The keyword that begins the part of an if statement that executes when none of the other conditions is true. The correct answer is: else

Quiz 4

Write a statement that **adds 1 to the value in the variable count**.  Use the **increment operator**. The correct answer is: count++;

Write a statement that **subtracts 1 from the value in the variable count**.  Use the **decrement operato**r. The correct answer is: count--;

Write the "opening line" of a **while loop** that could be used to **do something 10 times**.  Use the variable **count** in your loop condition.  Assume that count has been **declared and initialized to 1** and that the last statement in the body of the loop increments count. The correct answer is: while (count <= 10)

Write the "opening line" of a **for loop** that **declares an integer variable count** and **repeats as long as count is between 1 and 10** inclusive.  Don't include the opening curly brace!

The correct answer is: for (int count = 1; count <= 10; count++)

Write a statement that **adds 5 to the value in the variable count**.  DO NOT use the compound assignment operator. The correct answer is: count = count + 5;

Write a statement that **subtracts 1 from the value in the variable count**.  Use the **compound assignment operator**. The correct answer is: count -= 1;

Write the "opening line" of a **for** loop that **declares an integer variable countDown and initializes it to 10**.  The loop should **decrement countDown** with each iteration and should **repeat as long as countDown is bigger than 0**.  Don't include the opening curly brace!

The correct answer is: for (int countDown = 10; countDown >= 1; countDown--)

Write a statement that **adds 1 to the value in the variable count**.  Use the **compound assignment operator.** The correct answer is: count += 1;

Write the "opening line" of a **while loop** that could be used to validate that a variable number is divisible by 5.  The loop should **repeat as long as number is NOT divisible by 5**.  Assume that number has been declared an initialized prior to the loop and that the variable is updated as the last line of the loop body.

The correct answer is: while (number % 5 != 0)

Write the "opening line" of a **while** loop that will **repeat as long as the value in the variable isPrime is true**.  Assume that the variable isPrime has been declared and initialized before the loop begins. Don't include the opening curly brace.

The correct answer is: while (isPrime == true)

Write a statement that declares a string variable called name and assigns the results from reading a line from the keyboard to the variable.

The correct answer is: string name = Console.ReadLine();

Write a statement that declares a string variable called favoriteTeacher and assigns the results from reading a line from the keyboard to the variable.

string favoriteTeacher = Console.ReadLine();

Assume that you have 2 variables that can store integer values, feet and totalInches.  Write an assignment statement that calculates number of feet from the number of totalInches. Store the result in the variable feet.  27 inches should result in 2 feet.  45 inches should result in 3 feet.

feet = totalInches / 12;

wk 8 Quiz

* Write a statement that **declares** a real number variable **hours** and **assigns** it to the value returned from a **call** to the method **GetDouble**. The method takes **no** **parameters** and **returns** a **double**.
  + double hours = GetDouble();
* Write a statement that **sends the value** in the variable **prime** directly **back to the calling code** from inside a method;
  + return prime;
* Write the **header** of the method definition for a method named **GetDouble**. The method takes **no** **parameters** and **returns** a **double**.
  + double GetDouble()
* Write the **header** of the method definition for a method named **IsEven**. The method takes an **integer parameter called number**.  It **returns either true or false**.
  + bool IsEven(int number)
* Write a statement that **sends the value** in the variable **number** directly **back to the calling code** from inside a method;
  + return number;
* Write the **header** of the method definition for a method named **Roll2Dice**. The method takes **two integer parameters, d1 and d1.** Because it generates 2 values, it **does not directly return a value,** but returns the values in d1 and d2 indirectly through the use of **output parameters.**
  + void Roll2Dice(out int d1, out int d2)
* Write a statement that **declares a variable called even** and **assigns** it to the value returned from **calling** the method **IsEven** with a variable called **number** **(that has already been declared and initialized) as a parameter**. The method takes an **integer parameter and returns either true or false**.
  + bool even = IsEven(number);
* Assume that 2 integer variables **die1 and die2 have been declared**.  Write a statement that **calls** amethod named **Roll2Dice**. The method takes **two integer parameters.**Because it generates 2 values, the method **does not directly return a value,** but returns the values indirectly through the use of **output parameters.**
  + Roll2Dice(out die1, out die2);
* Write the **header** of the method definition for a method named **DisplayScores**. The method takes **two integer parameters, player1Score and player2Score.** It displays the scores on the screen but **does not return a value.**
  + void DisplayScores(int player1Score, int player2Score)
* Assume that 2 variables **userScore and computerScore have already been declared and initialized**. Write a statement that calls a method named **DisplayScores** to display the **userScore and computerScore** on the screen**.**The method takes **two integer parameters.** It displays the scores on the screen but **does not return a value.**
  + DisplayScores(userScore, computerScore);

Wk 9 Quiz 6

* Write a statement that **declares a variable words** and **initialize it** to the following list of words: **"one", "two", "three"**
  + string[] words = {"one", "two", "three"};
* Write a statement that **declares a variable scores** that can store an **array of up to 5 integers** and **allocates space** in memory for those 5 integers.
  + int[] scores = new int[5];
* Write a statement that **declares a variable words** that can store an **array of up to 10 strings**and **allocates space** in memory for those strings.
  + string[] words = new string[10];
* Write the **header** of the method definition for a method named **FillArray**. The method takes a **3 value parameters, a string called prompt and 2 integers called low and high**.  It **returns an array of integers**.
  + int[] FillArray(string prompt, int low, int high)
* Assume that **you have** an **array of integers called numbers** that has been declared and filled with values.  Write the **heading of a for loop** that can be used to iterate through the **entire** array.  Use **i** as the loop variable.
  + for (int i = 0; i < numbers.Length; i++)
* Write the **header** of the method definition for a method named **BubbleSort**. The method takes an **array of integers called numbers** and **an integer** representing the number of elements in the array **called count as parameters.** It**does not directly return a value,** but does alter the numbers array because an array is a reference type.
  + void BubbleSort(int[] numbers, int count)
* Write a statement that **declares an array of integers called lotteryPicks** and **assigns** it to the value **returned from calling a method named FillArray**. The method takes a **3 value parameters, a string called prompt and 2 integers called low and high.** Use the following literal values, **"lottery numbers", 1 and 69 as actual parameters** in the call.
  + int[] lotteryPicks = FillArray("lottery numbers", 1, 69);
* Assume that **you have** an **array of integers called numbers** that has been declared and filled with values.  Write a statement that **declares an integer variable called count** and **assigns it to the number of elements** in the array.
  + int count = numbers.Length;
* Assume that **you have** an **array of integers called numbers** that has been declared and filled with values.  Write a statement that **changes the first element** in the array to **10**;
  + numbers[0] = 10;
* Assume that **you have** an **array of integers called numbers** that has been declared and filled with values as well as an **integer count** that contains the number of elements in the array. Write a statement that **uses** a method named **BubbleSort** to sort the numbers array. The method takes an **array of integers** called numbers and **an integer** representing the number of elements in the array called count **as parameters.** The method**does not directly return a value,** but does alter the numbers array because an array is a reference type.
  + BubbleSort(numbers, count)