Congratulations! You passed!

Grade received 100% Latest Submission Grade 100% To pass 80% or higher

Go to next item

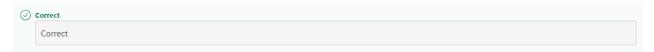
1/1 point

1. Fill in the blanks to print the even numbers from 2 to 12.

1 number = 2 # Initialize the variable
2 while number < 12+1: # Complete the while loop condition
3 print(number, end="")
4 number += 2 # Increment the variable
5
6 # Should print 2 4 6 8 10 12

Reset

2 4 6 8 10 12



2. Find and correct the error in the for loop. The loop should print every number from 5 to 0 in descending order.

1/1 point

```
for number in range(5,-1,-1):
       print(number)
   4
       # Should print:
   6
       # 4
       # 3
   8
      # 2
                                                                                                                              Run
       # 1
  10
       # 0
                                                                                                                             Reset
4
0
```



 $\textbf{3.} \quad \text{Fill in the blanks to complete the function "digits(n)" to count how many digits the given number has. For example: 25 has 2 digits and 144 has 3 digits.}$

1/1 point

Tip: you can count the digits of a number by dividing it by 10 once per digit until there are no digits left.

```
def digits(n):
             count = 0
if n == 0:
               count += 1
             while n >= 1: # Complete the while loop condition
                # Complete the body of the while loop. This should include
# performing a calculation and incrementing a variable in the
    6
                 # appropriate order.
                n = n/10
                 count += 1
           return count
   13
        print(digits(25)) # Should print 2
   14
         print(digits(144)) # Should print 3
   15
         print(digits(1000)) # Should print 4
                                                                                                                                                           Run
   16
         print(digits(0)) # Should print 1
                                                                                                                                                           Reset
4
1
```

1/1 point

4. Fill in the blanks to complete the "rows_asterisks" function. This function should print rows of asterisks (*), where the number of rows is equal to the "rows" variable. The number of asterisks per row should correspond to the row number (row 1 should have 1 asterisk, row 2 should have 2 asterisks, etc.). Complete the code so that "row_asterisks(5)" will print: **** * * * * * def rows_asterisks(rows): $\ensuremath{\text{\#}}$ Complete the outer loop range to control the number of rows for x in range(1, rows+1): # Complete the inner loop range to control the number of 4 # asterisks per row 5 for y in range(x): 6 # Prints one asterisk and one space print("*", end=" ") 8 # An empty print() function inserts a line break at the # end of the row 10 11 print() 12 13 rows_asterisks(5) 15 # Should print the asterisk rows shown above Run 16

5. Fill in the blanks to complete the "divisible" function. This function should count the number of values from 0 to the "max" parameter that are evenly divisible (no remainder) by the "divisor" parameter. Complete the code so that a function call like "divisible(100,10)" will return the number "10".

1/1 point

```
1
        def divisible(max, divisor):
            count = 1 # Initialize an incremental variable
            for x in range(divisor, \max): # Complete the for loop
              if x % divisor == 0:
                  count += 1 # Increment the appropriate variable
   6
            return count
        print(divisible(100, 10)) # Should be 10
                                                                                                                                          Run
        print(divisible(10, 3)) # Should be 4
   10
        print(divisible(144, 17)) # Should be 9
                                                                                                                                          Reset
10
9
```

6. Fill in the blanks to complete the "all_numbers" function. This function should return a space-separated string of all numbers, from the starting "minimum" variable up to and including the "maximum" variable that's passed into the function. Complete the for loop so that a function call like "all_numbers(3,6)" will return the numbers "3 4 5 6".

1/1 point

```
def all numbers(minimum, maximum):
         return_string = "" # Initializes variable as a string
         # Complete the for loop with a range that includes all
         # numbers up to and including the "maximum" value.
         for x in range(minimum, maximum+1):
q
             # Complete the body of the loop by appending the number
10
             # followed by a space to the "return_string" variable.
11
             return\_string += str(x) + "
12
         # This .strip command will remove the final " " space
13
         # at the end of the "return string".
14
```

```
return return_string.strip()
        15
        16
        17
        print(all_numbers(3,10)) # Should be 3 4 5 6 7 8 9 10
        19
        Run
                                                                                                                                      Reset
     2 3 4 5 6
3 4 5 6 7 8 9 10
     -1 0 1
     0 1 2 3 4 5
    ⊘ Correct
         Correct
7. The following code raises an error when executed. What's the reason for the error?
                                                                                                                                                     1/1 point
           def decade_counter():
              while year < 50:
year += 10
       3
              return year
   O Incrementing by 10 instead of 1
   O Nothing is happening inside the while loop

    Failure to initialize the variable

   O Wrong comparison operator
    ⊘ Correct
8. What is the final value of "x" at the end of this for loop? Your answer should be only one number.
                                                                                                                                                      1/1 point
           for x in range(1, 10, 3):
          print(x)
    ⊘ Correct
9. What number is printed at the end of this code?
                                                                                                                                                     1/1 point
           num1 = 0
           num2 = 0
           for x in range(5):
               for y in range(14):
                num2 = y + 3
           print(num1 + num2)
    20
    ⊘ Correct
```

```
def count_numbers(first, last):
    # Loop through the numbers from first to last
    x = first
    while x <= last:
    print(x)

count_numbers(2, 6)
    # Should print:
    # 2
    # 3
    # 4
    # 5
    # 5</pre>
```

O Wrong comparison operator is used

O Missing the break keyword

Missing an if-else block

Variable x is not incremented

