Congratulations! You passed!

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

Go to next item

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

1 / 1 point

```python
number = 2 # Initialize the variable
while number < 12+1: # Complete the while loop condition
    print(number, end=" ")
    number += 2 # Increment the variable

# Should print 2 4 6 8 10 12
```

Run
Reset

```
2 4 6 8 10 12
```

Correct

Correct

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

```python
for number in range(5,-1,-1):
    print(number)

# Should print:
# 5
# 4
# 3
# 2
# 1
# 0
```

Run
Reset

```
5
4
3
2
1
0
```

Correct

Correct

3. 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.

```python
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
        # appropriate order.
        n = n/10
        count += 1
    return count

print(digits(25))   # Should print 2
print(digits(144))  # Should print 3
print(digits(1000)) # Should print 4
print(digits(0))    # Should print 1
```

Run
Reset

```
2
3
4
1
```

Correct

Correct

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:

1 / 1 point

```
*

* *

* * *

* * * *

* * * * *
```

```python
def rows_asterisks(rows):
    # 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
        # asterisks per row
        for y in range(x):
        #     # Prints one asterisk and one space
            print("*", end=" ")
        # An empty print() function inserts a line break at the
        # end of the row
        print()


rows_asterisks(5)
# Should print the asterisk rows shown above
```

Run

Reset

```
*
* *
* * *
* * * *
* * * * *
```

Correct

Correct

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

```python
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
    return count

print(divisible(100, 10)) # Should be 10
print(divisible(10, 3)) # Should be 4
print(divisible(144, 17)) # Should be 9
```

Run

Reset

```
10
4
9
```

Correct

Correct

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

```python
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):

        # Complete the body of the loop by appending the number
        # followed by a space to the "return_string" variable.
        return_string += str(x) + " "

    # This .strip command will remove the final " " space
    # at the end of the "return string".
    return return_string.strip()


print(all_numbers(2,6))  # Should be 2 3 4 5 6
print(all_numbers(3,10)) # Should be 3 4 5 6 7 8 9 10
print(all_numbers(-1,1)) # Should be -1 0 1
print(all_numbers(0,5))  # Should be 0 1 2 3 4 5
print(all_numbers(0,0))  # Should be 0
```

Run

Reset

```
2 3 4 5 6
3 4 5 6 7 8 9 10
-1 0 1
0 1 2 3 4 5
0
```

Correct

Correct

7. The following code raises an error when executed. What's the reason for the error?

1 / 1 point

```python
def decade_counter():
    while year < 50:
        year += 10
    return year
```

- ○ Incrementing by 10 instead of 1
- ○ Nothing is happening inside the while loop
- [✓ Failure to initialize the variable]
- ○ 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

```python
for x in range(1, 10, 3):
    print(x)
```

7

Correct

9. What number is printed at the end of this code?

1 / 1 point

```python
num1 = 0
num2 = 0

for x in range(5):
    num1 = x
    for y in range(14):
        num2 = y + 3

print(num1 + num2)
```

20

Correct

10. The following code causes an infinite loop. Can you figure out what's missing and how to fix it?

1 / 1 point

```python
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
# 6
```

- Wrong comparison operator is used
- Missing the break keyword
- Missing an **if-else** block
- [✓ Variable x is not incremented]

Correct