Congratulations! You passed!

Grade received 100%

Latest Submission Grade 100%

To pass 80% or higher

Go to next item

Your computer's timezone does not seem to match your Coursera account's timezone setting of America/Los_Angeles.
Change your Coursera timezone setting

1. Fill in the blanks to print the numbers 1 through 7.

1 / 1 point

```python
number = 1 # Initialize the variable
while number < 8: # Complete the while loop condition
    print(number, end=" ")
    number += 1 # Increment the variable

# Should print 1 2 3 4 5 6 7
```

Run
Reset

```
1 2 3 4 5 6 7
```

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

Change your Coursera timezone setting

3. Fill in the blanks to complete the "factorial" function. This function will accept an integer variable "n" through the function parameters and produce the factorials of this number (by multiplying this value by every number less than the original number [n*(n-1)], excluding 0). To do this, the function should:

1 / 1 point

- accept an integer variable "n" through the function parameters;
- initialize a variable "result" to the value of the "n" variable;
- iterate over the values of "n" using a while loop until "n" is equal to 0;
- starting at n-1, multiply the result by the current "n" value;
- decrement "n" by -1.

For example, factorial 3 would return the value of 3*2*1, which would be 6.

```python
def factorial(n):
    result = n
    start = n
    n -= 1
    while n > 0: # The while loop should execute as long as n is greater than 0
        result *= n # Multiply the current result by the current value of n
        n -=1 # Decrement the appropriate variable by -1
    return result


print(factorial(3)) # Should print 6
print(factorial(9)) # Should print 362880
print(factorial(1)) # Should print 1
```

Run
Reset

```
6
362880
1
```

Correct

Correct

Change your Coursera timezone setting

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(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=" ")
```

Change your Coursera timezone setting

```
            # 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 “counter” function. This function should count down from the “start” to “stop” variables when “start” is bigger than “stop”. Otherwise, it should count up from “start” to “stop”. Complete the code so that a function call like “counter(3, 1)” will return “Counting down: 3, 2, 1” and “counter(2, 5)” will return “Counting up: 2, 3, 4, 5”.

1 / 1 point

```
def counter(start, stop):
    if start > stop:
        return_string = "Counting down: "
        while stop != start + 1: # Complete the while loop
            return_string += str(start) # Add the numbers to the "return_string"
            if start > stop:
                return_string += ","
            start -= 1 # Increment the appropriate variable
    else:
        return_string = "Counting up: "
        while start != stop + 1 : # Complete the while loop
            return_string += str(start) # Add the numbers to the "return_string"
            if start < stop:
                return_string += ","
            start +=1 # Increment the appropriate variable
    return return_string


print(counter(1, 10)) # Should be "Counting up: 1,2,3,4,5,6,7,8,9,10"
print(counter(2, 1)) # Should be "Counting down: 2,1"
print(counter(5, 5)) # Should be "Counting up: 5"
```

Run

Reset

```
Counting up: 1,2,3,4,5,6,7,8,9,10
Counting down: 2,1
Counting up: 5
```

✓ **Correct**

Correct

Change your Coursera timezone setting

6. Fill in the blanks to complete the “even_numbers” function. This function should return a space-separated string of all positive even numbers, excluding 0, up to and including the “maximum” variable that's passed into the function. Complete the for loop so that a function call like “even_numbers(6)” will return the numbers “2 4 6”.

1 / 1 point

```
def even_numbers(maximum):

    return_string = "" # Initializes variable as a string

    # Complete the for loop with a range that includes all even numbers
    # up to and including the "maximum" value, but excluding 0.
    for num in range(1 ,maximum + 1):
        if num % 2 ==0:
            return_string += str(num) + " "
        # Complete the body of the loop by appending the even number
        # followed by a space to the "return_string" variable.

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

print(even_numbers(6))  # Should be 2 4 6
print(even_numbers(10)) # Should be 2 4 6 8 10
print(even_numbers(1))  # No numbers displayed
print(even_numbers(3))  # Should be 2
print(even_numbers(0))  # No numbers displayed
```

Run

Reset

```
2 4 6
2 4 6 8 10

2

```

✓ **Correct**

Correct

Change your Coursera timezone setting

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
    return year
```

● Failure to initialize the variable

○ Incrementing by 10 instead of 1

○ Nothing is happening inside the while loop

○ Wrong comparison operator

✓ **Correct**

Change your Coursera timezone setting

8. How many numbers will this loop print? Your answer should be only one number.

1 / 1 point

```
for sum in range(5):
    sum += sum
    print(sum)
```

5

✓ Correct

Change your Coursera timezone setting

9. What is the initial value of the "outer_loop" variable on the first iteration of the nested "inner_loop"? Your answer should be only one number.

1 / 1 point

```
for outer_loop in range(2, 6+1):
    for inner_loop in range(outer_loop):
        if inner_loop % 2 == 0:
            print(inner_loop)
```

Change your Coursera timezone setting

2

✓ Correct

10. The following code causes an infinite loop. Can you figure out what is incorrect?

1 / 1 point

```
def test_code(num):
  x = num
  while x % 2 == 0:
    x = x / 2

test_code(0)
```

○ Missing an **else** statement

○ The modulo operator is used incorrectly

○ Missing the **continue** keyword

● When called with 0, it triggers an infinite loop

✓ Correct