



Module 3 Graded Assessment

TOTAL POINTS 10

1. Fill in the blanks of this code to print out the numbers 1 through 7.

1 point

```
1 number = 1
2 while number <= 7:
3     print(number, end=" ")
4     number += 1
```

1 2 3 4 5 6 7

2. The show_letters function should print out each letter of a word on a separate line. Fill in the blanks to make that happen.

1 point

```
1 def show_letters(word):
2     for letter in word:
3         print(letter)
4
5 show_letters("Hello")
6 # Should print one line per letter
```

H
e
l
l
o

3. Complete the function digits(n) that returns how many digits the number has. For example: 25 has 2 digits and 144 has 3 digits. **Tip:** you can figure out the digits of a number by dividing it by 10 once per digit until there are no digits left.

1 point

```
1 def digits(n):
2     count = 0
3     if n == 0:
4         count = 1
5     while (n > 0):
6         count += 1
7         n = int(n / 10)
8     return count
9
10 print(digits(25)) # Should print 2
11 print(digits(144)) # Should print 3
12 print(digits(1000)) # Should print 4
13 print(digits(0)) # Should print 1
```

2
3
4
1

4. This function prints out a multiplication table (where each number is the result of multiplying the first number of its row by the number at the top of its column). Fill in the blanks so that calling multiplication_table(1, 3) will print out:

1 point

1 2 3

2 4 6

3 6 9

```
1 def multiplication_table(start, stop):
2     for x in range(start, stop+1):
3         for y in range(start, stop+1):
4             print(str(x*y), end=" ")
5         print()
6
7 multiplication_table(1, 3)
8 # Should print the multiplication table shown above
```

1 2 3
2 4 6
3 6 9

5. The counter function counts down from start to stop when start is bigger than stop, and counts up from start to stop otherwise. Fill in the blanks to make this work correctly.

1 point

```
1 def counter(start, stop):
2     x = start
3     if start > stop:
4         return string = "Counting down: "
```

```

4   return_string = "Counting down: "
5   while x >= stop:
6       return_string += str(x)
7       if x > stop:
8           return_string += ","
9       x -= 1
10  else:
11      return_string = "Counting up: "
12      while x <= stop:
13          return_string += str(x)
14          if x < stop:
15              return_string += ","
16          x += 1
17  return return_string
18
19 print(counter(1, 10)) # Should be "Counting up: 1,2,3,4,5,6,7,8,9,10"
20 print(counter(2, 1)) # Should be "Counting down: 2,1"
21 print(counter(5, 5)) # Should be "Counting up: 5"

```

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

6. The even_numbers function returns a space-separated string of all positive numbers that are divisible by 2, up to and including the maximum that's passed into the function. For example, even_numbers(6) returns "2 4 6". Fill in the blank to make this work.

```

1  def even_numbers(maximum):
2      return_string = ""
3      for x in range(1, maximum+1):
4          if x % 2 == 0: return_string += str(x) + " "
5      return return_string.strip()
6
7  print(even_numbers(6)) # Should be 2 4 6
8  print(even_numbers(10)) # Should be 2 4 6 8 10
9  print(even_numbers(1)) # No numbers displayed
10 print(even_numbers(3)) # Should be 2
11 print(even_numbers(0)) # No numbers displayed

```

2 4 6
2 4 6 8 10
2

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

```

1  def decade_counter():
2      while year < 50:
3          year += 10
4      return year

```

- ☐ Incrementing by 10 instead of 1
- ☒ Failure to initialize variables
- ☐ Nothing is happening inside the while loop
- ☐ Wrong comparison operator

8. What is the value of x at the end of the following code?

```

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

```

7

9. What is the value of y at the end of the following code?

```

1  for x in range(10):
2      for y in range(x):
3          print(y)

```

8

10. How does this function need to be called to print yes, no, and maybe as possible options to vote for?

```

4

```

☐ votes("yes", "no", "maybe")

- ☐ votes(yes, no, maybe)
- ☐ votes([yes, no, maybe])
- ☒ votes(['yes', 'no', 'maybe'])

☒ I, **Piyush Sambhi**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

[Learn more about Coursera's Honor Code](#)



Save

Submit