1. What is a **function** in Python? And why should we use them?

2. Label each listed item in the following program:

1. Function Header
2. Loop Variable
3. Function Body
4. Accumulator Variable
5. Docstring
6. Function Definition
7. Parameter(s)
8. Function Call
9. Global Variable
10. Return Statement
11. Local Variable

```python
import random
def say_something(something):
    '''this function prints something'''
    to_print = 'You asked for it, so here is: '+something+'.
    print(to_print)
    to_return = random.randrange(0,101)
    return to_return

a = "apple"
i = 0
for letter in a:
    i += say_something(letter)
print(i)
```
3. Does it matter whether you end your function with `return x` or `print(x)`?

4. Consider the following program:

```python
def times_three(x):
    return x * 3
x = 'three'
x = 3.333
x = int(x)
x = times_three(x)
```

There are multiple variables named `x` in this program, what caused that? Is this a good idea?

5. This function uses the accumulator pattern to do a common mathematical computation. What does it compute?

```python
def mystery(num1, num2):
    num3 = 1
    for count in range(num2):
        num3 *= num1
    return num3
```

6. Will all of the lines in the following program be executed?

```python
def a_function():
    print('this function')
a = "doesn't really"
return "do anything"
print("now does it?")
```
def happy(day):
    return "I'm happy because today is " + day + "!

def sad(day):
    return "I'm sad because today is not" + day + "... :("

favorite_day = 'Friday'
for current_day in ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday']:
    if current_day == favorite_day:
        print( current_day + ":", happy(current_day) )
    else:
        print( current_day + ":", sad(current_day) )

def add(num, x):
    return num + x

def subtract(num, x):
    return num - x

count = 0
for num in range(4):
    if num % 2 == 0:
        count = add(num, count)
    else:
        count = subtract(count, num)
9. What is the output of this program? If there are any errors, identify and correct them.

```python
def funct1(a, b, c):
    print(b)
    print(a)
    print(c + a)
    return a + b

def funct2(a, b, c):
    print(b)
    return c + a

def funct3(a, b, c):
    print("hi, my name is", b)
    return b + a

a = "apple"
b = "baby"
c = "cat"
d = funct1(b, c, a)
e = funct2(d, a, b)
print( funct3(a, e, funct1(e, d, c) ) )
```

10. What is a main function and why would you use one?

11. What is returned from a function when we do not write a return statement?
12. What will the following function return with the given inputs:

```python
def test_function(x):
    y = ""
    for letter in str(x):
        y += letter + str(x)
    return(y)
```

1. "dog"
2. 'computer'
3. 12.43

13. What code is executed first when this module is run?

```python
def a_function(a):
    return a + 3

def b_function(b, x):
    return (b + x) / 2

def main():
    a = input("Give me a number: ")
    x = a_function(a)
    b = input("Give me a number: ")
    x = b_function(b, x)
    print("The result is", x)

main()
```

14. What would the output of the above program be with the inputs 2 and 7? If there are any errors in the code, correct them.
15. Suppose there is a function called `isThereClass(date)` which takes a string date (e.g., "September 20, 2017") and returns `True` if you have COSC 101 class on that date and `False` if you do not. Write a program that asks the user for the date and uses this function to determine whether or not they have class. Your program should print the result in a user-friendly sentence.

16. You are working at a big software company, everyone on your team is working together to finish a program. You have been asked to write a function called `display_date()` that takes three int parameters, `month`, `day`, and `year` and displays the date in a readable format. For example, `display_date(12, 313, 2017)` prints `December 31, 2017`. Write this function: