1. Consider this code:

```
def sum(number1, number2):
    print(number1 + number2)

>>result = sum(1, 3)
>>new_result = result + 1
```

What is the outcome of executing the code above?

a) new_result refers to the value 5
b) TypeError

2. What is printed by the code below?

```
def sum(number1, number2):
    return number1 + number2
    print("hello")

>>result = sum(1, 3)
```

a) “hello”
b) Nothing

3. Consider this function call:

```
print("this \n is the newline character in Python")
```

What is printed by this function call?

a) "this \n is the newline character in Python"
b) this \n is the newline character in Python
c) "this
is the newline character in Python"
d) this
is the newline character in Python

4. The function `count_vowels` returns the number of vowels in a given word (a `str`). For example:

```
>>> count_vowels("moose")
3
>>> count_vowels("scratch")
1
```

The **Type Contract** is a comment that describes the types of the parameters and return value, where the input type is first and is followed by the output type. Select the correct Type Contract for this function.

a) (str) -> int
b) (int) -> str
c) (str) -> float
d) (bool) -> str

5. We talked a lot about unit testing in class, including how best to unit test certain functions types. Give at least one way you could unit test the function below (ie. a written explanation for logic, not actual code).

```
def UpdateMulti(x,y,z,i):
    #x, z are floats, y is a string, i is an int
    #This function takes the arguments x,y,z and
    #updates them following a preset plan, given by the
    # fourth argument i
```

```
If i==0:
     x += 6
     y = “Hello!”
     z /= 2
     return
elif i==1:
     x -= 6
     y = “Goodbye!”
     z *= 2
     return
else:
     print(“Invalid plan option”)
     return
```