CS 8
Quiz 2: Exception Handling, Functions, and Modules

Name: **SOLUTION**

1. When should you use `TypeError`? How should you use it? (3 points)

When a function receives an argument whose type is not the expected kind, the function should raise an exception. Such as a function getting a string, but expecting an int or float.

2. When should you raise an exception? Give one example or situation. (3 points)

When writing a function (or method) and the function encounters an invalid situation. I would raise an exception to indicate the invalid situation. For an example, see the answer to #1.

3. What is the output of the following code snippet? Show all output in the box on the right. If the code crashes, indicate where the crash occurs. (2 points)

```
print('a')
try:
    print('b')
    raise ValueError
    print('c')
except TypeError:
    print('d')
except Exception:
    print('e')
print('f')
```

```
a
b
e
f
```

4. What is one of the reasons to create a function? (2 points)

There are three main reasons:

1. Divide and conquer – divide a program into smaller pieces and implement each piece
2. Reuse code – write code once and use it in multiple places (without having to copy and paste)
3. Functional abstraction or hiding implementation details – the user doesn't need to worry about *how* something works, just how to call the function so that it works

5. Why write a function that has parameters (as opposed to one without parameters)? (3 points)

To allow the user to customize how the function works (e.g. some of the optional parameters in the print function). Maybe the function needs certain values to work at all (e.g. comparing objects for equality). There are a lot of correct answers here.

6. What is `sys.path`? Why might you use it? (1 point)

It's a list of paths (directories) where Python modules are stored. One might use it to tell Python where some additional Python modules are stored.

7. What is the output of the following code snippet? Show all output in the box on the right. If the code crashes, indicate where the crash occurs. (3 points)

```
print('a')
def print_val(val):
    print('b')
    print(val)
    return 'c'
    print('d')

print('e')
print(print_val('f'))
print('g')
```

```
a
e
b
f
c
g
```

8. What is the output of the following code snippet? Show all output in the box on the right. If the code crashes, indicate where the crash occurs. (3 points)

```
def functionA():
    print('a')
    raise ValueError
    print('b')

def functionB():
    print('c')
    try:
        functionA()
        print('d')
    except TypeError:
        print('e')
    print('f')

print('g')
try:
    functionB()
    print('h')
except ValueError:
    print('i')
```

```
g
c
a
i
```