Name:_____ Section:___ Andrew Id: _____

15-112 Fall 2019 Quiz 1a

*** Up to 30 minutes. No calculators, no notes, no books, no computers. * Show your work!**

*** No strings, lists, string or list indexing, loops, or recursion**

1. **Multiple Choice** [10 pts]: Write your answers neatly in each box.

1) Which of the following would fill in the blank so that the code exhibits short-circuit evaluation:

```
x = 42
if ((x > 0) _________________ (x**2 > 100)): print('yes!')
```

A. and
B. or
C. xor
D. None of these

2) What is a variable's "scope"?

A. The name of the variable.
B. Where the variable can be seen or used.
C. The type of value the variable can refer to.
D. The largest value the variable can refer to.

3) Which of the following is NOT a category of errors a program can have, according to our class notes?

A. Syntax Errors
B. Runtime Errors
C. Logical Errors
D. Short-Circuit Evaluation Errors

4) Why do we suggest you use roundHalfUp(x) instead of round(x)?

A. Because round(x) sometimes crashes.
B. Because round(x) truncates instead of rounds.
C. Because round(x) rounds in a surprising way sometimes.
D. None of these

5) What is the main difference between a statement and an expression?

A. An expression can call a function, but a statement cannot.
B. An expression evaluates to a value, but a statement does not.
C. An expression can be included in a function, but a statement cannot.
D. None of these

2. **Free Response: largestPerfectSquare(n)** [10 pts]
   Write the function largestPerfectSquare(n) that takes a non-negative int n, and returns the largest perfect square that is no larger than n. For example:

```
assert(largestPerfectSquare(24) == 16)
assert(largestPerfectSquare(25) == 25)
assert(largestPerfectSquare(26) == 25)
```

   Hint: you may wish to use a similar approach to how you solved isPerfectSquare on the hw.
   Another hint: This can be written using just one or two lines of Python.

3. **Free Response: isFactorish(n)** [40 pts]
   Write the function isFactorish(n) that takes a value n that can be of any type, and returns True if n is a (possibly-negative) integer with exactly 3 unique digits (so no two digits are the same), where each of the digits is a factor of the number n itself. In all other cases, the function returns False (without crashing).
   For example:

```
assert(isFactorish(412) == True)      # 4, 1, and 2 are all factors of 412
assert(isFactorish(-412) == True)     # Must work for negative numbers!
assert(isFactorish(4128) == False)    # 4128 has more than 3 digits
assert(isFactorish(112) == False)     # 112 has duplicate digits (two 1's)
assert(isFactorish(420) == False)     # 420 has a 0 (no 0's allowed)
assert(isFactorish(42) == False)      # 42 has a leading 0 (no 0's allowed)
assert(isFactorish(412.0) == False)   # 412.0 is not an int
assert(isFactorish('nope!') == False) # don't crash on strings
```

4. **Code Tracing** [20 pts]:Indicate what these print. Place your answers (and nothing else) in the boxes below the code.

```
def ct1(x, y):
    print((x//10) % ((y%10)**3))
    if (x > y):
        return isinstance(x/10, type(x))

print(ct1(137,42))
print(ct1(42, 731))
```

```
def f(z):
    return 2*z

def g(z):
    z += 1
    return z/2

def h(z):
    if (z > 3):
        return z + f(g(z))
    else:
        return g(z)

def ct2(z):
    print(h(z-1))
    z *= 2
    return h(z)

print(ct2(3))
```

5. **Reasoning Over Code** [20 pts]:
Find arguments for the following functions that makes them return True. Place your answers (and nothing else) in the boxes below the code:

```
def rc1(n):
    t = n//1000
    return ((n >= 100000) and
            (n <= 333333) and
            (t != 112) and
            (t % 112 == 0) and
            (t == n%1000))
```

n =

```
def f(x1, x2, n):
    d1 = (x1 // (10**n)) % 10
    d2 = (x2 // (10**n)) % 10
    if ((d1 > d2) and (d1 > 5)):
        return d1
    elif (d2 > d1):
        return d2
    elif ((d1 == 0) and (d2 == 0)):
        return 42
    elif ((d1 == 0) or (d2 == 0)):
        return -10**10
    else:
        return 0

def rc2(x, y):
    z = 100*f(x,y,2) + 10*f(x,y,1) + f(x,y,0)
    return ((f(x,y,3) == 42) and
            (z == 206))
```

x =
y =

6. **Bonus/Optional: Code Tracing** [2.5 pts]
   Indicate what this prints. Place your answer (and nothing else) in the box below the code):

```
def f(x): return x+5
def g(x): return f(x-3)
def h(x): return g(g(x)%f(x))
def bonusCt1(f, g, x):
    if (x > 0):
        return bonusCt1(g, h, -f(x))
    else:
        return f(g(h(x)))
print(bonusCt1(g, f, 4))
```

7. **Bonus/Optional: Reasoning Over Code** [2.5 pts]
   Find an argument for the following function that makes it return True. Place your answer (and nothing else) in the box below the code):

```
def bonusRc1(x):
    assert(isinstance(x, int))
    def f(x): return ((x+x//x)**2 - (x-x**0)**2)
    return (f(f(f(x))) - f(x) == 360)
```

x =