## Control Flow and Repeat answer key

1.

a) when you need the user to enter an unknown number of input.

b) Could make cleaner code. Instead of loading up on conditions in the header of the loop you can write conditions in the body of the loop to gracefully exit.

c) For GUI and server programming. Many event handler loops have the program in a cyclic infinite loop waiting for an external event to occur.

2

a) With a valid condition and break statement.

b) With a KeyboardInterrupt exception.

c) CTRL – C

3. If you’re prototyping a program and have functionality that you’ll implement later.

4. No.

5. Yes.

>>> True and not True

False

>>> True and not False

True

>>> False and not True

False

>>> False and not False

False

6. Yes.

a) True o’red with anything is always true.

>>> True or not True

True

>>> True or not False

True

b) Yes.

>>> not (True and not True)

True

>>> not (False and not False)

True

c) Yes. This statement translated into English means the double negation of P implies P.

>>> not not True

True

>>> not not False

False

d) Yes. Order doesn’t matter. (True or False) is logically equivalent to (False or True).

e) (True and False) is logically equivalent to (False and True).

7. In code snippet *a* after the condition verifies to true the control of the program flows past the else statement. In code snippet *b*, it’s three if statements which will all be checked regardless of how the other conditions evaluate. In code snippet *b,* x==5 evaluates to true so x is printed, and x >=5 is also true so x is printed once again.

8.

a) It’s bad. A classic example of a quadratic algorithm is bubble sort which is ok for sorting small sets of numbers but has significant performance penalties when you’re trying to sort large lists of numbers.

b) You should see that the inner loop updates a lot more often that the outer loop.

9.

>>> def fizz\_buzz(n):

... """fizz buzz implementation in python"""

... if n < 1 or n > 100:

... print('ERROR')

... elif n % 3 == 0 and n % 5 == 0:

... print('FizzBuzz')

... elif n % 3 == 0:

... print('Fizz')

... elif n % 5 == 0:

... print('Buzz')

... else:

... print(n)

...

>>> fizz\_buzz(1)

1

>>> fizz\_buzz(6)

Fizz

>>> fizz\_buzz(20)

Buzz

>>> fizz\_buzz(15)

FizzBuzz

10.

>>> def check(n):

... if n < 1 or n > 100:

... print('ERROR')

... elif n >= 90 and n <=99:

... print('nada')

... elif n % 2 == 0:

... print('normal')

... elif n % 2 == 1:

... print('weird')

...

>>> check(0)

ERROR

>>> check(100)

normal

>>> check(99)

nada

>>> check(3)