**Practice Quiz 3: Conditionals**

**1.What's the value of this Python expression: (2\*\*2) == 4?**

True

**2.Complete the script by filling in the missing parts. The function receives a name, then returns a greeting based on whether or not that name is "Taylor".**

def greeting(name):

if name == "Taylor":

return "Welcome back Taylor!"

else:

return "Hello there, " + name

print(greeting("Taylor"))

print(greeting("John"))

**3.What’s the output of this code if number equals 10?**

if number > 11:

print(0)

elif number != 10:

print(1)

elif number >= 20 or number < 12:

print(2)

else:

print(3)

2

**4.Is "A dog" smaller or larger than "A mouse"? Is 9999+8888 smaller or larger than 100\*100? Replace the plus sign in the following code to let Python check it for you and then answer.print("A dog" < "A mouse") print(9999+8888 > 100\*100)**

"A dog" is smaller than "A mouse" and 9999+8888 is larger than 100\*100

**5.If a filesystem has a block size of 4096 bytes, this means that a file comprised of only one byte will still use 4096 bytes of storage. A file made up of 4097 bytes will use 4096\*2=8192 bytes of storage. Knowing this, can you fill in the gaps in the calculate\_storage function below, which calculates the total number of bytes needed to store a file of a given size?**

def calculate\_storage(filesize):

block\_size = 4096

# Use floor division to calculate how many blocks are fully occupied

full\_blocks = filesize // block\_size

# Use the modulo operator to check whether there's any remainder

partial\_block\_remainder = filesize % block\_size

# Depending on whether there's a remainder or not, return

# the total number of bytes required to allocate enough blocks

# to store your data.

if partial\_block\_remainder > 0:

return (full\_blocks + 1) \* block\_size

return full\_blocks \* block\_size

print(calculate\_storage(1)) # Should be 4096

print(calculate\_storage(4096)) # Should be 4096

print(calculate\_storage(4097)) # Should be 8192

print(calculate\_storage(6000)) # Should be 8192