1) . What is the difference between enclosing a list comprehension in square brackets and

parentheses?

A: square Brackets are used to make lists . Parenthesis are used to make tuple

2) What is the relationship between generators and iterators?

A: **ITERATOR** is an object which contains a countable number of values and it is used to iterate over iterable objects like list, tuples, sets, etc. Iterators are implemented using a class and a local variable for iterating is not required here, It follows lazy evaluation where the evaluation of the expression will be on hold and stored in the memory until the item is called specifically which helps us to avoid repeated evaluation.

**GENERATOR:** another way of creating iterators in a simple way where it uses the keyword “yield” instead of returning it in a defined function. Generators are implemented using a function.

3) What are the signs that a function is a generator function?

A: def generator\_name(arg):

# statements

yield something

4) What is the purpose of a yield statement?

A: The yield statement returns a generator object to the one who calls the function which contains yield, instead of simply returning a value

5) What is the relationship between map calls and list comprehensions? Make a comparison and

contrast between the two.

A: List comprehension returns a list, whereas the map function returns an object of iterable

A: def num (n) :

return n \* 2

lst = [2, 44, 5.5, 6, -7]

# suppose we want to call function

# 'num' for each element of lst,

# we use map

# creates a map object

x = map(num, lst)

print(x)

# returns list

print(list(x))