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

A. Square brackets are used to create a list comprehension, while parentheses are used to create a generator expression.

2) What is the relationship between generators and iterators?

A. All generators are iterator, but not all iterators are generators. Generators are a specific type of iterator that are defined using a generator function and use the yield statement to generate values on-the-fly.

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

A. The signs that a function is a generator function are the use of the “yield” keyword instead of “return”, the return of a generator object when the function is called, and the ability to iterate over the generator object using the “next()” function to produce the next value in the sequence.

4) What is the purpose of a yield statement?

A. The “yield” statement is used in a generator function to temporarily suspend execution and return a value to the caller, allowing the function to produce a sequence of values on-the-fly. This can be useful for producing potentially infinite sequences, or for working with large datasets that cannot be stored in memory all at once.

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

A. Map calls and list comprehensions are both used in python to transform data from one format to another, but they have some differences in terms of syntax and functionality. Map calls are generally more flexible, but require conversion to a list or other iterable before they can be used, whereas list comprehensions are limited to expressions but return a list object directly.