Python Viva Questions – Detailed Notes

## What are Python’s key features?

Python is an open-source, high-level, interpreted, dynamically typed language with support for multiple programming paradigms including procedural, object-oriented, and functional programming. It emphasizes code readability and has a vast standard library.

## Difference between list, tuple, and set?

- List: Mutable, ordered collection that allows duplicate elements.  
- Tuple: Immutable, ordered collection that allows duplicates.  
- Set: Mutable, unordered collection that does not allow duplicates.

## Explain Python’s memory management.

Python uses reference counting as well as a cyclic garbage collector to manage memory automatically. Each object has a reference count that tracks the number of references to it. When it reaches zero, the memory is released.

## What are \*args and \*\*kwargs?

\*args allows a function to accept any number of positional arguments as a tuple. \*\*kwargs allows a function to accept any number of keyword arguments as a dictionary.

## What is a lambda function?

A lambda function is a small anonymous function defined with the `lambda` keyword. It can take any number of arguments but has only one expression.  
Example: lambda x: x + 2

## What are decorators in Python?

Decorators are functions that modify the behavior of other functions or methods. They are used with the `@` symbol and are often used for logging, access control, instrumentation, etc.

## What is the difference between `is` and `==`?

`is` checks for object identity (whether two references point to the same object in memory), while `==` checks for value equality (whether the objects referred to have the same value).

## What are Python’s basic data types?

Python’s core data types include int, float, str, bool, list, tuple, dict, and set. Each of these types supports various operations and methods.

## Difference between modules and packages?

- Module: A single .py file containing Python definitions and statements.  
- Package: A directory containing multiple modules and an `\_\_init\_\_.py` file to mark it as a package.

## Explain exception handling in Python.

Python handles exceptions using try-except blocks. You can also use finally for code that must run no matter what, and else for code that runs only if no exception occurs.  
Example:  
```python  
try:  
 x = 10 / 0  
except ZeroDivisionError:  
 print('Cannot divide by zero')  
finally:  
 print('Done')  
```