

PRE LAB:-

1.Differences Between Shallow Copy and Deep Copy in Python?

- **Shallow Copy:** Creates a new object but inserts references into it to the objects found in the original. Changes to mutable objects within the copied object will affect the original. Use `copy.copy()` for shallow copying.
- **Deep Copy:** Creates a new object and recursively copies all objects found in the original, resulting in a completely independent copy. Use `copy.deepcopy()` for deep copying.

2.Usage of Lambda Functions in Python?

Lambda functions are anonymous functions defined using the `lambda` keyword, typically for short, throwaway functions.

Example:

```
add = lambda x, y: x + y
```

```
print(add(3, 5)) # Output: 8
```

3.Python Virtual Environment?

A Python virtual environment is an isolated environment in which you can install packages independently from the system-wide Python installation. It's used to manage dependencies for different projects, preventing conflicts and ensuring consistency.

4.Differences Between == and is Operators?

== Operator: Checks for value equality, meaning it compares if the values of two objects are the same.

is Operator: Checks for identity equality, meaning it compares if two references point to the same object in memory.

5.Role of self in Python Class Methods:?

The `self` parameter in a class method refers to the instance of the class, allowing access to instance attributes and methods. It must be the first parameter in instance methods to distinguish between instance and class attributes.

VIVA :-

1.Python Module vs. Package?

- **Module:** A module is a single file containing Python code, such as functions, classes, and variables, that can be imported using `import`. Example: `math` module.
- **Package:** A package is a collection of modules organized in directories with an `__init__.py` file, allowing for a hierarchical structure of modules. Example: `numpy` package.

2.Handling Exceptions in Python?

Exceptions in Python are handled using `try`, `except`, `else`, and `finally` blocks.

Example:

```
try:
    result = 10 / 0
except ZeroDivisionError:
    print("Cannot divide by zero.")
else:
    print("Division successful.")
finally:
    print("Execution completed.")
```

3. Purpose of the `__init__` Method in Python Classes?

The `__init__` method is a special method called a constructor, used to initialize a new object's attributes when an instance of the class is created. It allows for setting up initial values for the object.

4. Differences Between `range()` and `xrange()` in Python 2?

- **`range()`**: Returns a list of numbers from start to stop (inclusive of start, exclusive of stop) in Python 2.
- **`xrange()`**: Returns an iterator generating numbers on demand, which is more memory efficient for large ranges. `xrange()` is not available in Python 3, where `range()` behaves like `xrange()`.

5. Decorators in Python?

Decorators are functions that modify or enhance other functions or methods. They are used to add functionality to existing code in a clean and reusable way.