INSTITUTE OF TECHNOLOGY AND MANAGEMENT
SKILLS UNIVERSITY,
KHARGHAR, NAVI MUMBAI

PYTHON PROGRAMMING LAB

Prepared by:

Name of Student: _Shikha singh_____

Roll No: _25_____

Batch: 2023-27

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Exp. No	List of Experiment
	1.1 Write a program to compute Simple Interest.
	1.2 Write a program to perform arithmetic, Relational operators.
	1.3 Write a program to find whether a given no is even & odd.
	1.4 Write a program to print first n natural number & their sum.
	1.5 Write a program to determine whether the character entered is a Vowel or not .
	1.6 Write a program to find whether given number is an Armstrong Number.
	1.7 Write a program using for loop to calculate factorial of a No.
	1.8 Write a program to print the following pattern
	i) * * * * * * * * * * * * * * *
	ii) 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
	iii) *

	2.1 Write a program that define the list of defines the list of define countries that are in BRICS.
	2.2 Write a program to traverse a list in reverse order. 1.By using Reverse method. 2.By using slicing
	2.3 Write a program that scans the email address and forms a tuple of username and domain.
	2.4 Write a program to create a list of tuples from given list having number and add its cube in tuple. i/p: c= [2,3,4,5,6,7,8,9]
	2.5 Write a program to compare two dictionaries in Python? (By using == operator)
	2.6 Write a program that creates dictionary of cube of odd numbers in the range.
	2.7 Write a program for various list slicing operation. a= [10,20,30,40,50,60,70,80,90,100] i. Print Complete list ii. Print 4th element of list iii. Print list from0th to 4th index. iv. Print list -7th to 3rd element v. Appending an element to list. vi. Sorting the element of list. vii. Popping an element. viii. Removing Specified element. ix. Entering an element at specified index. x. Counting the occurrence of a specified element. xi. Extending list. xii. Reversing the list.
	3.1 Write a program to extend a list in python by using given approach. i. By using + operator. ii. By using Append ()

	iii. By using extend ()
	3.2 Write a program to add two matrices.
	3.3 Write a Python function that takes a list and returns a new list with distinct elements from the first list.
	3.4 Write a program to Check whether a number is perfect or not.
	3.5 Write a Python function that accepts a string and counts the number of upper- and lower-case letters. string_test= 'Today is My Best Day'
	4.1 Write a program to Create Employee Class & add methods to get employee details & print.
	4.2 Write a program to take input as name, email & age from user using combination of keywords argument and positional arguments (*args and**kwargs) using function,
	4.3 Write a program to admit the students in the different Departments(pgdm/btech) and count the students. (Class, Object and Constructor).
	4.4 Write a program that has a class store which keeps the record of code and price of product display the menu of all product and prompt to enter the quantity of each item required and finally generate the bill and display the total amount.
	4.5 Write a program to take input from user for addition of two numbers using (single inheritance).
	4.6 Write a program to create two base classes LU and ITM and one derived class. (Multiple inheritance).
	4.7 Write a program to implement Multilevel inheritance, Grandfather□Father-□Child to show property inheritance from grandfather to child.
	4.8 Write a program Design the Library catalogue system using inheritance take base class (library item) and derived class (Book, DVD & Journal) Each derived

	class should have unique attribute and methods and system should support Check in and check out the system. (Using Inheritance and Method overriding)
	5.1 Write a program to create my_module for addition of two numbers and import it in main script.
	5.2 Write a program to create the Bank Module to perform the operations such as Check the Balance, withdraw and deposit the money in bank account and import the module in main file.
	5.3 Write a program to create a package with name cars and add different modules (such as BMW, AUDI, NISSAN) having classes and functionality and import them in main file cars.
	6.1 Write a program to implement Multithreading. Printing “Hello” with one thread & printing “Hi” with another thread.
.	7.1 Write a program to use ‘whether API’ and print temperature of any city, also print the sunrise and sunset times for the same humidity of that area.
	7.2 Write a program to use the ‘API’ of crypto currency.

Name of Student: _Shikha singh_____

Roll Number: _25_____

Experiment No: 4.7

Title:

4.7 Write a program to implement Multilevel inheritance, GrandfatherFather-Child to show property inheritance from grandfather to child.

Theory:

Inheritance is one of the fundamental concepts of object-oriented programming (OOP) that allows a class to inherit properties and behaviors from another class. Multilevel inheritance occurs when a class is derived from another class, and then another class is derived from that derived class. In the context of your request, we have the following hierarchy:

- **Grandfather is the base class.**
- **Father inherits from Grandfather.**
- **Child inherits from Father.**

This creates a chain of inheritance, where the Child class indirectly inherits properties and behaviors from both the Father and Grandfather classes.
Key Points:

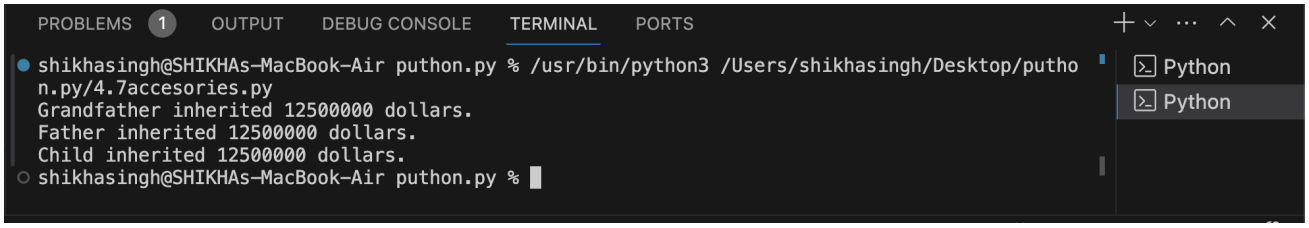
- **Base Class (Grandfather): The class from which properties and behaviors are inherited. It serves as the foundation for the entire hierarchy.**
- **Derived Class (Father and Child): Classes that inherit from other classes. They can extend the functionality of the base class and add their own unique properties and methods.**
- **Inheritance Chain: In multilevel inheritance, there is a chain of classes, and each class inherits from the one above it in the hierarchy.**

Code:

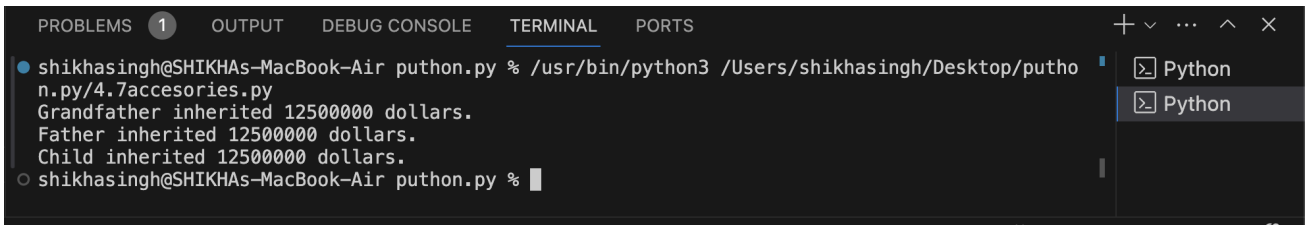
```
class Grandfather:
    def __init__(self, money_amount):
        self.money = money_amount
        print(f"Grandfather inherited {self.money} dollars.")
class Father(Grandfather):
    def __init__(self, money_amount):
        super().__init__(money_amount // 2)
        print(f"Father inherited {self.money} dollars.")
class Child(Father):
    def __init__(self, money_amount):
        super().__init__(money_amount // 2)
        print(f"Child inherited {self.money} dollars.")
def main():

    grandfather_money = 50000000
    my_child = Child(grandfather_money)
if __name__ == "__main__":
    main()
```

Output: (screenshot)



Test Case: Any two (screenshot)



Conclusion:

Multilevel inheritance is a powerful feature in OOP that allows for the creation of complex class hierarchies. It promotes code reusability, as properties and behaviors defined in the base class can be inherited by multiple derived classes. However, it's important to use inheritance judiciously to avoid creating overly complex and tightly coupled class hierarchies.

In the provided example, the Child class inherits properties from both the Father and Grandfather classes, showcasing how properties are inherited through multiple levels of the hierarchy. This concept is fundamental in building organized and efficient class structures in object-oriented programming.