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	1.1 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	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	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	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	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	6.1 Write a program to implement Multithreading. Printing “Hello” with one thread & printing “Hi” with another thread.
7.	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:

4.3

Title: 4.3 Write a program to admit the students in the different Departments(pgdm/btech)and count the students. (Class, Object and Constructor).

Theory: **Class ITM:**

- **The ITM class is designed to represent student information at an institution (presumably named ITM).**
- **It includes class variables (count, bcount, pcount) to keep track of the overall admissions, BTECH admissions, and PGDM admissions.**

- **Constructor (__init__):**
- **The class has an __init__ method to initialize instance variables (count, bcount, pcount) for each object of the class.**

- **Methods (getData and setData):**
- **getData method collects information from the user about a student's name, age, address, and department (BTECH or PGDM).**
- **The method updates class variables based on the department chosen and increments the overall admission count.**
- **setData method displays the collected student details.**

- **Main Program:**
- **The main program initializes an empty list (objs) to store instances of the ITM class.**
- **It presents a menu-driven interface for the user to choose operations like entering student data, checking the number of admissions, and displaying student details.**
- **The program uses a while loop to continuously execute the chosen operations until the user chooses to exit.**

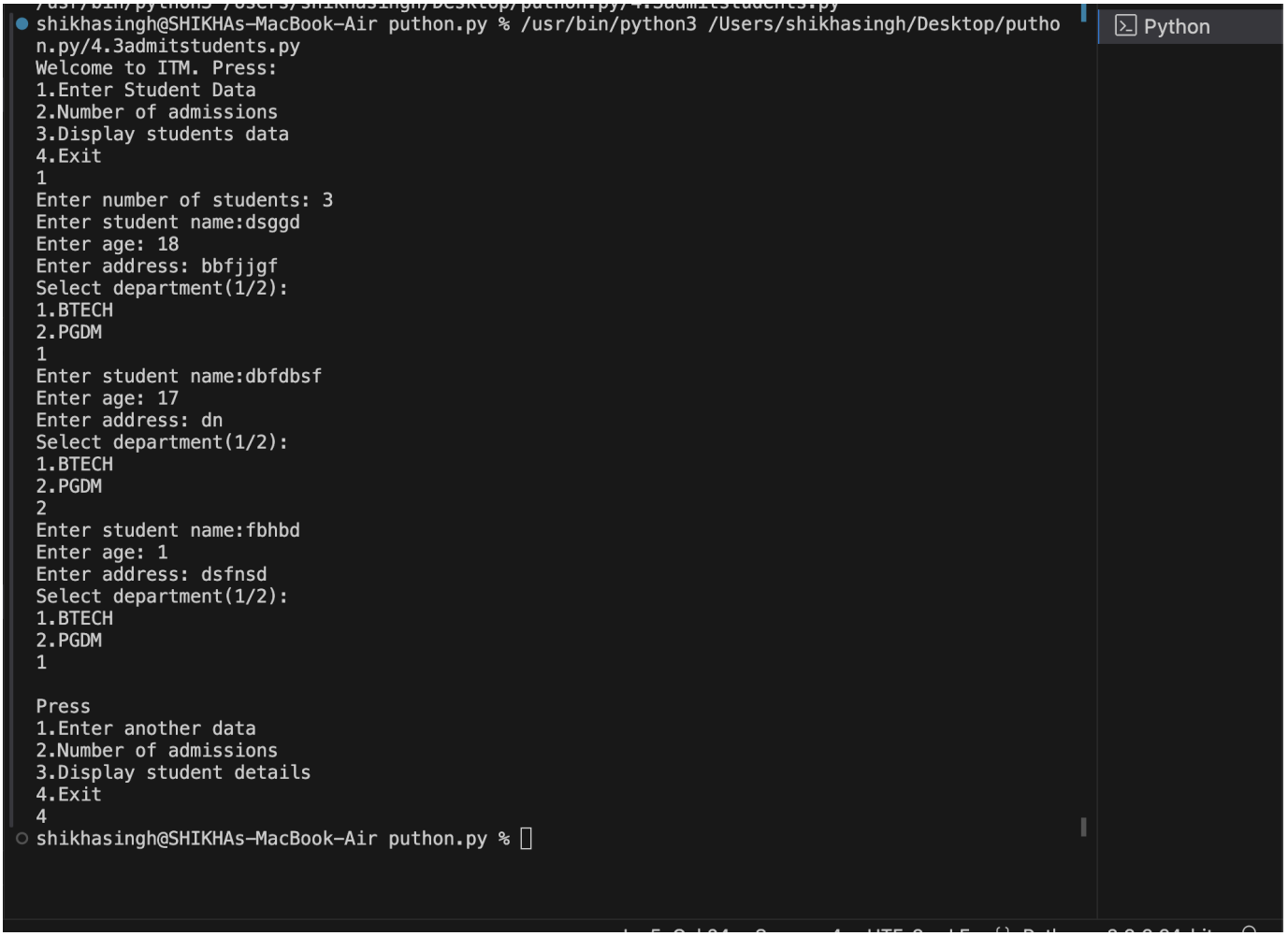
Code: **class ITM:**

```
count=0
bcount=0
pcount=0
def __init__(self):
   self.count=0
   self.bcount=0
   self.pcount=0
def getData(self):
   a=input("Enter student name:" )
   b=int(input("Enter age: "))
   c=input("Enter address: ")
   d=int(input("Select department(1/2):\n1.BTECH\n2.PGDM\n"))
   while not(d==1 or d==2):
      print("Invalid choice\n")
      d=int(input("Select department(1/2):\n1.BTECH\n2.PGDM\n"))
   if d==1:
      self.dep="BTECH"
      ITM.bcount+=1
   elif d==2:
      self.dep="PGDM"
      ITM.pcount+=1
```

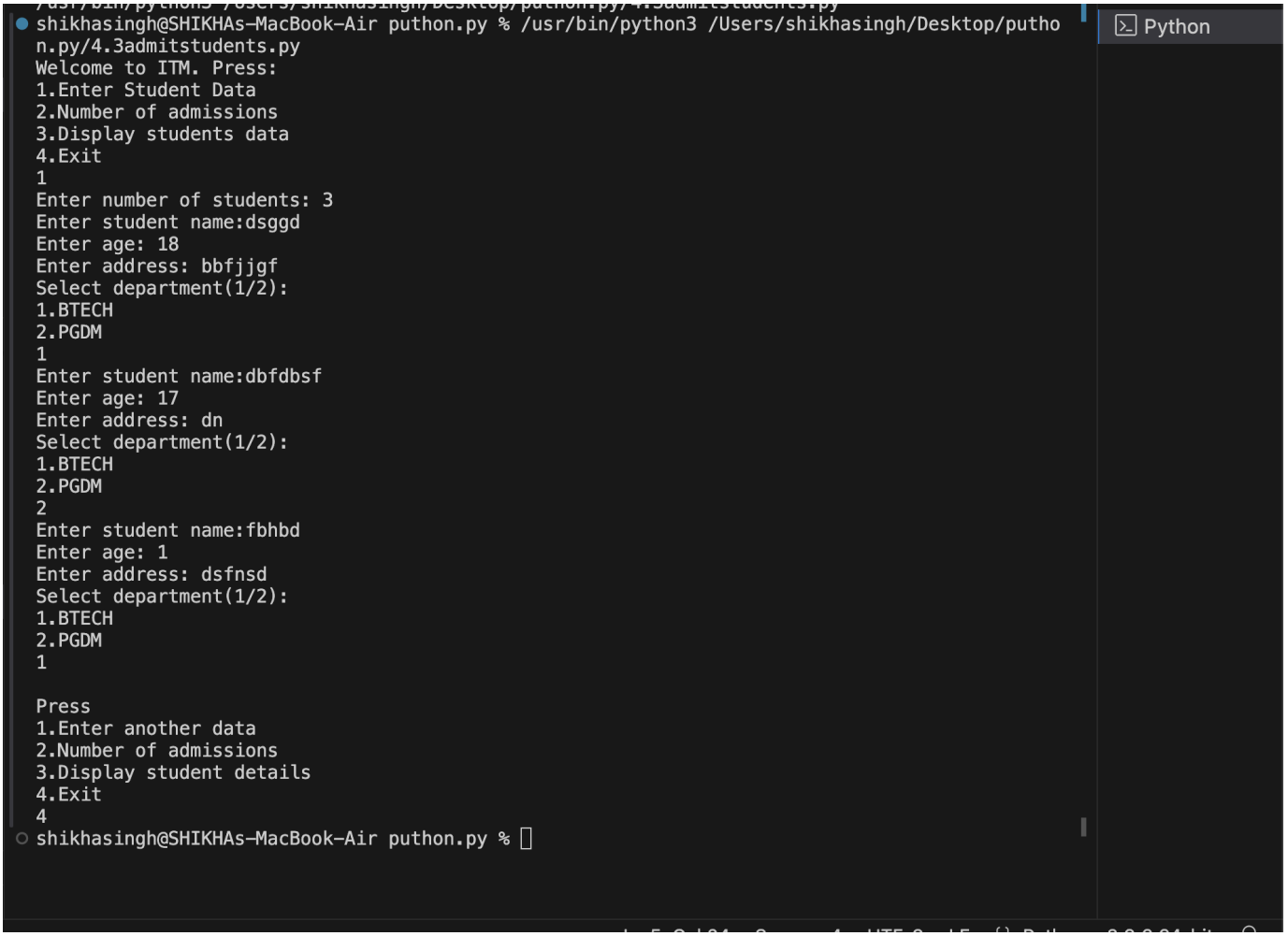
```
        self.name=a
        self.age=b
        self.address=c
        ITM.count+=1
    def setData(self):
        print("Student Details:\n")
        print("Name:",self.name)
        print("\nAge:",self.age)
        print("\nAddress:",self.address)
        print("\nDepartment:",self.dep)
a=int(input("Welcome to ITM. Press:\n1.Enter Student Data\n2.Number of admissions\n3.Display students
data\n4.Exit\n"))
objs=list()
while a!=4:
    if a==1:
        d=int(input("Enter number of students: "))
        for i in range(d):
            objs.append(ITM())
        for i in range(d):
            objs[i].getData()
        a=int(input("\nPress\n1.Enter another data\n2.Number of admissions\n3.Display student details\n4.Exit\n"))
    elif a==2:
        b=int(input("Press\n1.BTECH admissions\n2.PGDM admissions\n"))
        while not (b==1 or b==2):
            print("Invalid choice\n")
            b=int(input("Press\n1.BTECH admissions\n2.PGDM admissions\n"))
        if b==1:
            print("Admissions of BTECH Done:",ITM.bcount)
            print("Total admissions:",ITM.count)
            a=int(input("\nPress\n1.Enter student data\n2.Number of admissions\n3.Display student details\n4.Exit\n"))
        elif b==2:
            print("Admissions of PGDM Done:",ITM.pcount)
            print("Total admissions:",ITM.count)
            a=int(input("\nPress\n1.Enter student data\n2.Number of admissions\n3.Display student details\n4.Exit\n"))
    elif a==3:
        b=int(input("\nPress\n1.For BTECH students data\n2.For PGDM students data\n"))
        if b==1:
            for i in range(d):
                if objs[i].dep=="BTECH":
                    objs[i].setData()
            a=int(input("\nPress\n1.Enter student data\n2.Number of admissions\n3.Display student details\n4.Exit\n"))
        elif b==2:
            for i in range(d):
                if objs[i].dep=="PGDM":
                    objs[i].setData()
            a=int(input("\nPress\n1.Enter student data\n2.Number of admissions\n3.Display student details\n4.Exit\n"))
        else:
            print("Invalid choice")
            a=int(input("\nPress\n1.Enter student data\n2.Number of admissions\n3.Display student details\n4.Exit\n"))
    elif a==4:
        break
```

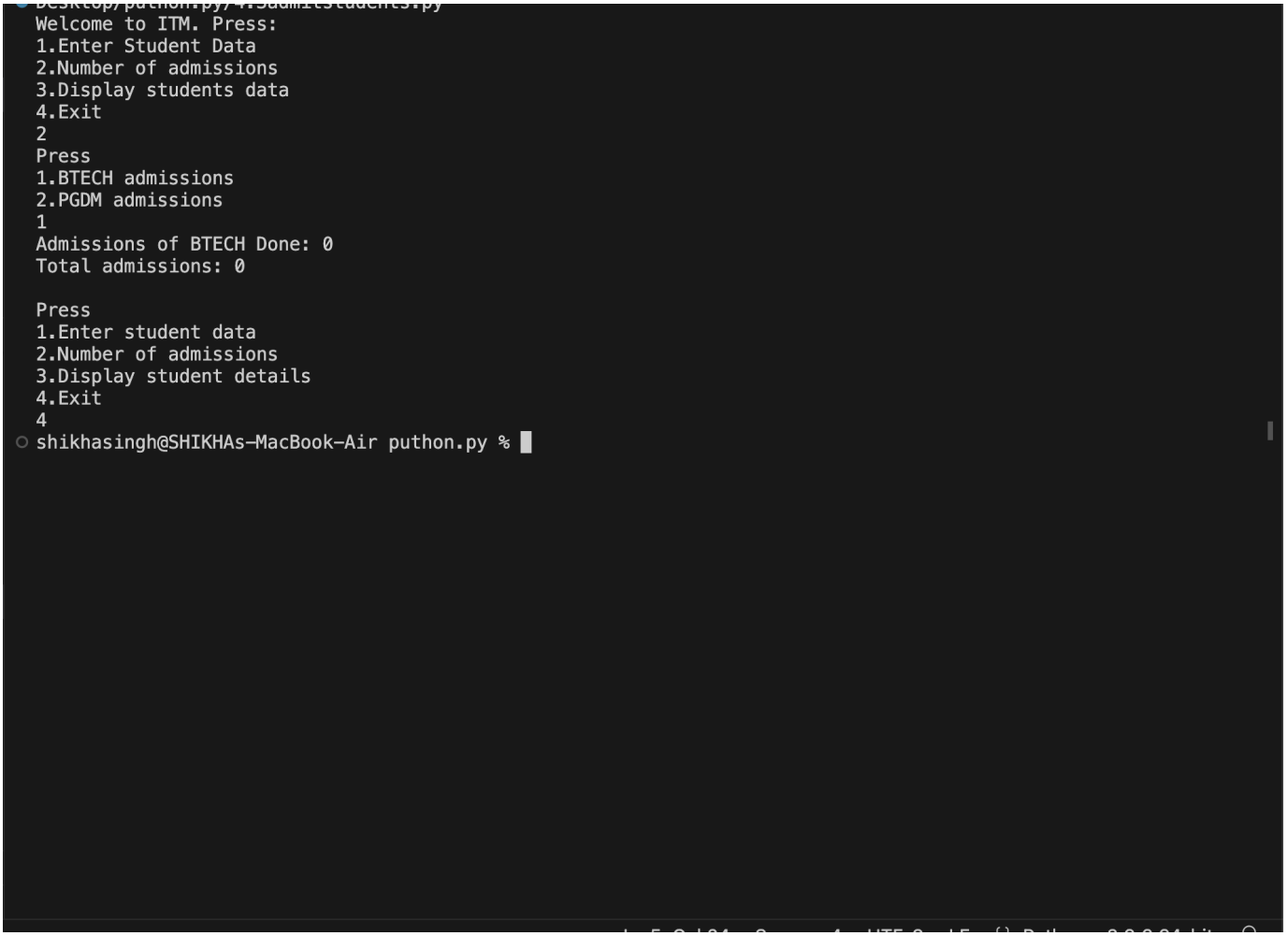
```
else:
    print("Invalid choice")
    break
```

Output: (screenshot)



Test Case: Any two (screenshot)


Python
shikhasingh@SHIKHAs-MacBook-Air python.py % /usr/bin/python3 /Users/shikhasingh/Desktop/putho
n.py/4.3admitstudents.py
Welcome to ITM. Press:
1.Enter Student Data
2.Number of admissions
3.Display students data
4.Exit
1
Enter number of students: 3
Enter student name:dsggd
Enter age: 18
Enter address: bbfjjgf
Select department(1/2):
1.BTECH
2.PGDM
1
Enter student name:dbfdbsf
Enter age: 17
Enter address: dn
Select department(1/2):
1.BTECH
2.PGDM
2
Enter student name:fbhbd
Enter age: 1
Enter address: dsfnsd
Select department(1/2):
1.BTECH
2.PGDM
1
Press
1.Enter another data
2.Number of admissions
3.Display student details
4.Exit
4
shikhasingh@SHIKHAs-MacBook-Air python.py %



Conclusion:**The program provides a**

basic interface for managing student

admissions and displaying student details at

the ITM institution.

- **It uses object-oriented programming concepts with a class (ITM) to encapsulate student-related functionalities.**
- **The program handles user input and validates choices to ensure the correct execution of operations.**
- **By using a list (objs) to store instances of the ITM class, the program can manage multiple student objects.**
- **The modular design allows for easy extension and modification of functionalities.**
- **However, there could be improvements, such as adding error handling, better organization of code, and enhancing the user interface for a more user-friendly experience.**