Vidyalankar Polytechnic

DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	1
Title of Experiment	Install and configure Python IDE

X. PRACTICAL RELATED QUESTIONS

1. Write steps for installing Python on window.

ANS:

a) Visit the official website python.org/downloads and click on the latest Python version download button for Windows
b) Run the downloaded installer file and make sure to check the box that says "Add Python to PATH" at the bottom of the installer window
c) Click "Install Now" to start installation with recommended settings (or choose "Customize installation" if you need specific components)
d) Wait for the installation progress bar to complete - this usually takes 1-2 minutes depending on your system
e) Test the installation by opening Command Prompt (cmd) and typing "python --version" - if you see a version number displayed, Python is successfully installed

2. State IDLE in Python

Ans:

a) It's Python's default Integrated Development Environment
b) Provides syntax highlighting for better code readability
c) Includes an interactive Python shell (interpreter)
d) Has a simple text editor with basic debugging tools
e) Perfect for beginners learning Python programming

3. List key features of Python

Ans:

- Simple to learn and use
- Open Source
- High Level Language
- DynamicallyTyped
- Platform Independent
- Portable
- Procedure and Object Oriented

4. Explain Python Path

Ans:

a) PYTHONPATH is an environment variable
b) Tells Python where to find module files for importing
c) Search order: Current directory → PYTHONPATH → Default path
d) Can be modified to include custom module directories
e) Essential for managing project dependencies

5. State use of pep and pip

Ans:

PEP (Python Enhancement Proposals):

- Design documents for Python features
- Provides coding standards (like PEP 8)
- Ensures consistency in Python community
- Describes Python's processes and guidelines
- Helps maintain code quality

pip (Package Installer for Python):

- Command-line tool for installing Python packages
- Manages package dependencies
- Updates and removes packages
- Creates requirements.txt files
- Integrates with virtual environments

XI. Exercise

1. Print the version of Python

Ans:

```
import platform

print(platform.python_version())
```

2. Write steps to be followed to load Python interpreter in windows.

Ans:

a) Click on the desktop, click on the start button, then click on Run

b) On your desktop:
   - ➢ Click on the Start button
   - ➢ Click on Programs
   - ➢ Click on Python 2.8

c) The third method involves running the Python interpreter:
   - ➢ Execute by typing its file name at an MSDOS shell command line

	DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	2
Title of Experiment	Write simple Python program to display message on screen

X. PRACTICAL RELATED QUESTIONS

1. List different modes of Programming in Python.

ANS:

a) Script
b) Interactive

2. Describe procedure to execute program using Interactive Mode

Ans:

The Interactive Mode:

a) Involves running code directly in the Python shell

b) Can be accessed from the terminal of the operating system

c) In script mode, you must:

- Create a file
- Give it a .py extension
- Run your code

d) Interactive mode is suitable when writing a few lines of code

3. State the steps involved in executing the program using Script Mode

Ans:

a) File > New File or Press Ctrl+N
b) Write the Python code as function/script
c) Save it (.py)
d) Execute it in interactive mode using RUN option (F5)

4. State the procedure to make file executable

Ans:

Terminal Execution Steps:

a) Open a terminal

b) Browse to the folder where the executable file is stored

c) Type the command:

   - For any .bin file: sudo chmod +x filename.bin

   - For any .run file: sudo chmod +x filename.run

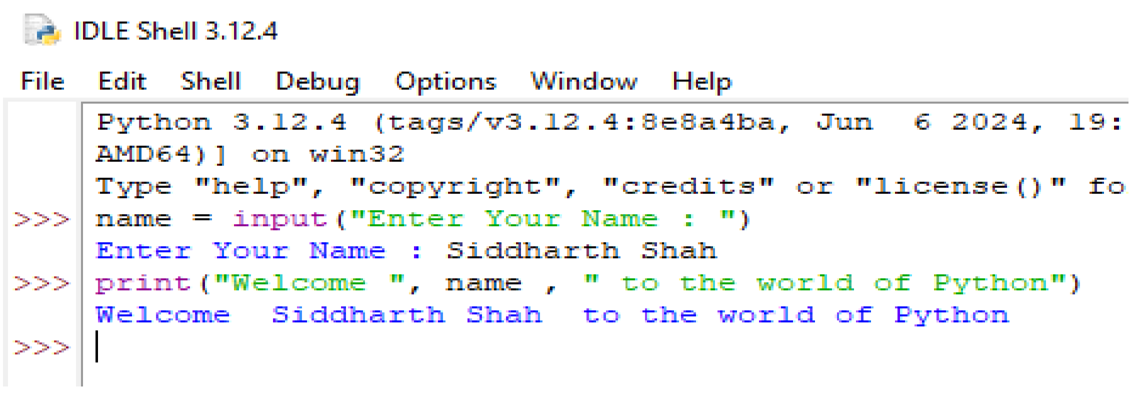
d) When asked, type the required password and press Enter

XI. Exercise

1. Write a Python program to display your name using Interactive Mode.

Ans:

a] Output:



```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" fo
>>> name = input("Enter Your Name : ")
Enter Your Name : Siddharth Shah
>>> print("Welcome ", name , " to the world of Python")
Welcome  Siddharth Shah  to the world of Python
>>> 
```

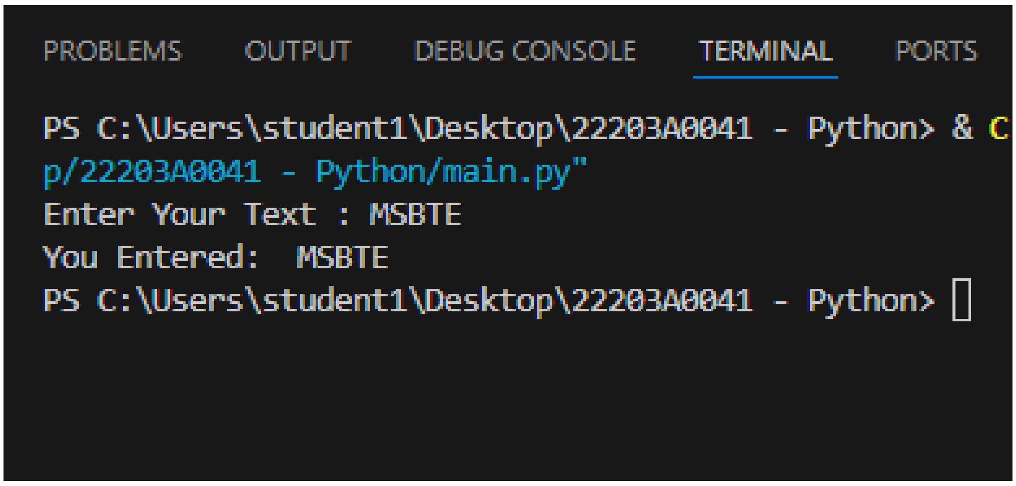
2. Write a Python program to display “MSBTE” using Script Mode.

Ans:

a] Code:

```
name = input("Enter Your Text : ")

print("You Entered: " , name)
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C
p/22203A0041 - Python/main.py"
Enter Your Text : MSBTE
You Entered:  MSBTE
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

Vidyalankar Polytechnic	DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	3
Title of Experiment	Write simple Python program using operators: Arithmetic Operators, Logical Operators, Bitwise Operators

X. PRACTICAL RELATED QUESTIONS

1. Mention the use of //, **, % operator in Python.

ANS:

a] Code:

i] //

```
a = 10

b = 5

print("Floor Division : " , (a//b))
```

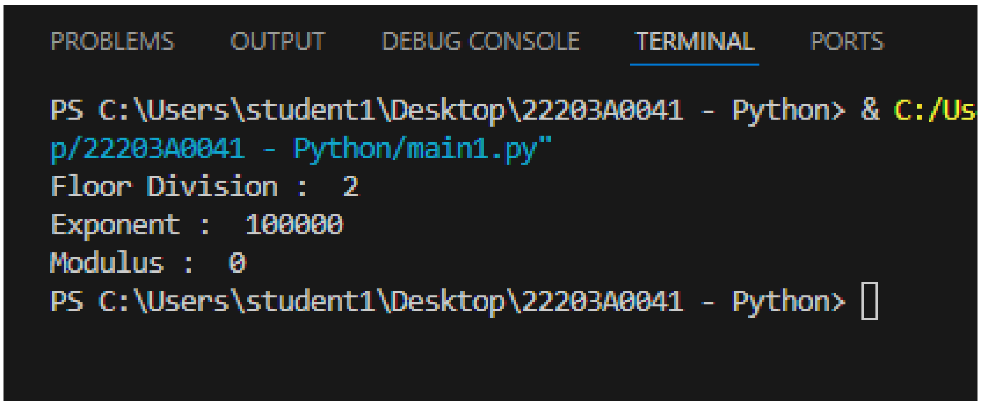
ii] **

```
a = 10

b = 5

print("Exponent : " , (a**b))
```

iii] %

```
a = 10

b = 5

print("Modulus : " , (a%b))
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/Us
p/22203A0041 - Python/main1.py"
Floor Division :  2
Exponent :  100000
Modulus :  0
PS C:\Users\student1\Desktop\22203A0041 - Python>
```

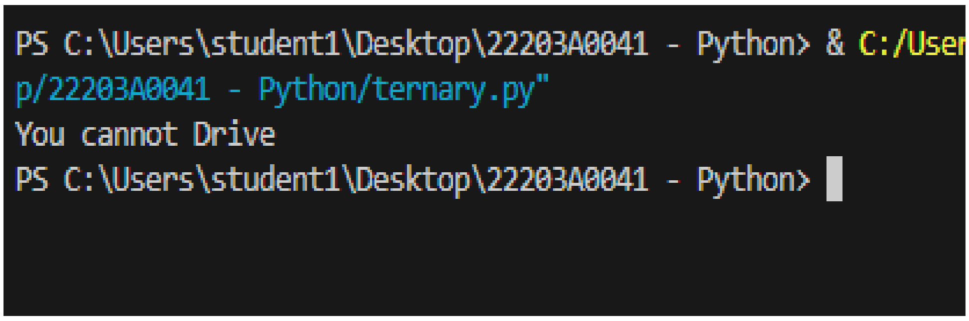
2. Describe ternary operator in Python

Ans:

a] Code:

```
age = 16

result = "You can drive" if age>18 else "You cannot Drive"

print(result)
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/User
p/22203A0041 - Python/ternary.py"
You cannot Drive
PS C:\Users\student1\Desktop\22203A0041 - Python>
```

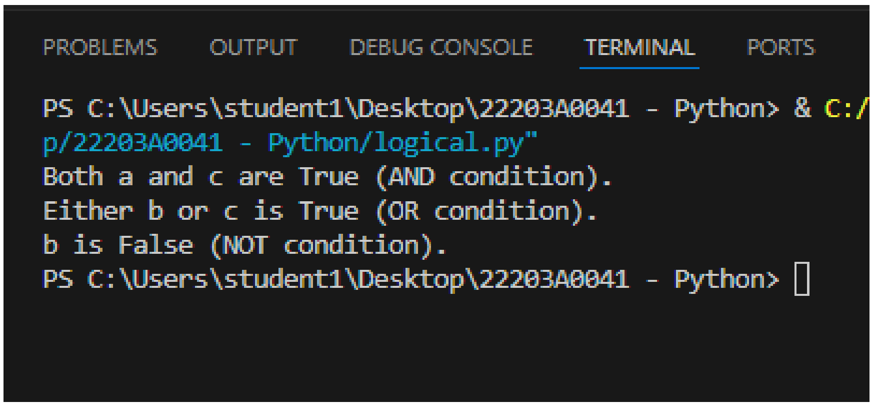
3. Describe about different Logical operators in Python with appropriate examples.

Ans:

a] Code:

```
a = True

b = False

c = True

if a and c:

    print("Both a and c are True (AND condition).")

if b or c:

    print("Either b or c is True (OR condition).")

if not b:

    print("b is False (NOT condition).")
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/
p/22203A0041 - Python/logical.py"
Both a and c are True (AND condition).
Either b or c is True (OR condition).
b is False (NOT condition).
PS C:\Users\student1\Desktop\22203A0041 - Python>
```

4. Describe about different Arithmetic operators in Python with appropriate examples

Ans:

a] Code:

```
num1=40

num2=10

print("number 1 :", num1)

print("number 2 :", num2)
```

print("Addition of " , num1 , " + " , num2 , " = " , (num1+num2))

print("Subtraction of " , num1 , " - " , num2 , " = " , (num1-num2))

print("Multiplication of " , num1 , " * " , num2 , " = " , (num1*num2))

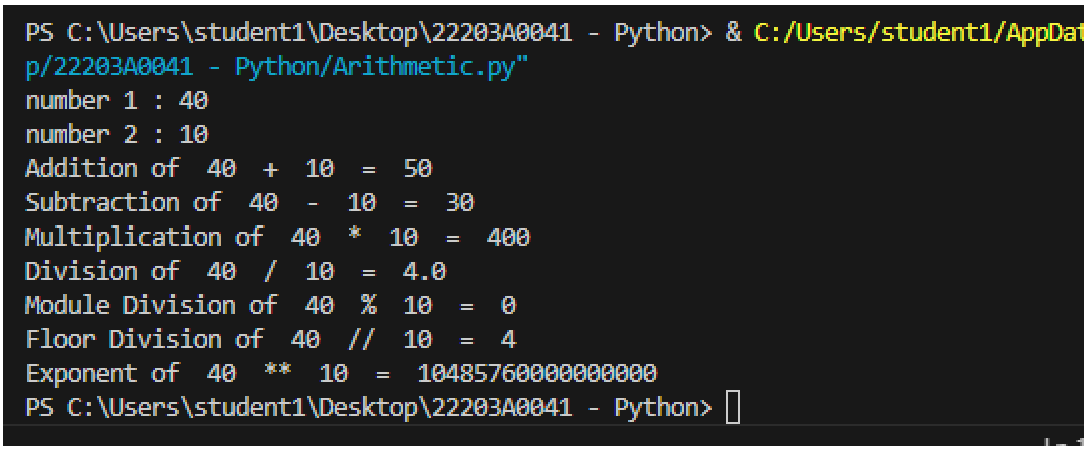
print("Division of " , num1 , " / " , num2 , " = " , (num1/num2))

print("Module Division of " , num1 , " % " , num2 , " = " , (num1%num2))

print("Floor Division of " , num1 , " // " , num2 , " = " , (num1//num2))

print("Exponent of " , num1 , " ** " , num2 , " = " , (num1**num2))

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/Users/student1/AppDat
p/22203A0041 - Python/Arithmetic.py"
number 1 : 40
number 2 : 10
Addition of  40  +  10  =  50
Subtraction of  40  -  10  =  30
Multiplication of  40  *  10  =  400
Division of  40  /  10  =  4.0
Module Division of  40  %  10  =  0
Floor Division of  40  //  10  =  4
Exponent of  40  **  10  =  10485760000000000
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

5. Describe about different Arithmetic operators in Python with appropriate examples

Ans:

a] Code:

role = 0b1001

hr = 0b1011

admin = 0b1001

manager = 0b1100

if(hr & role):

    print("You are HR")

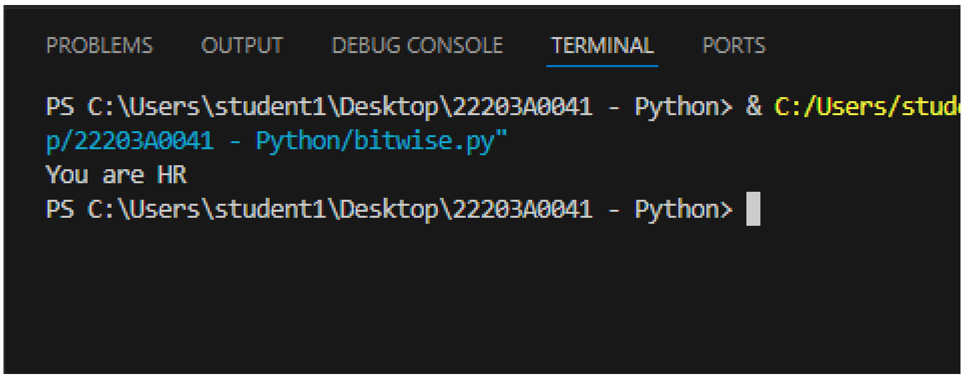
elif(admin & role):

    print("your role is Admin")

else:

```
print("Your role is Manager")
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/Users/stud
p/22203A0041 - Python/bitwise.py"
You are HR
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

XI. Exercise

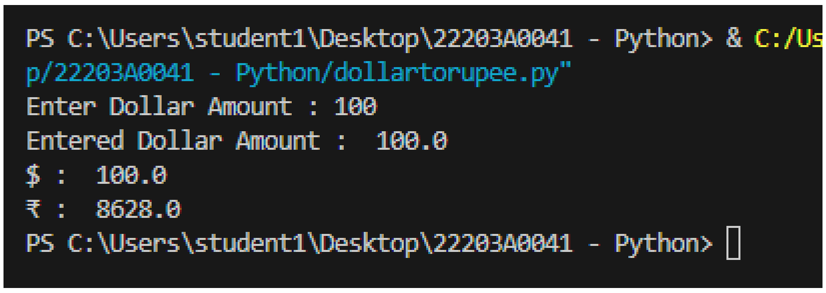
1. Write a program to convert U.S. dollars to Indian rupees.

Ans:

a] Code:

```
dollar = float(input("Enter Dollar Amount : "))

print("Entered Dollar Amount : " , dollar)

print("$ : " , dollar)

print("₹ : " , dollar*86.28)
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/Us
p/22203A0041 - Python/dollartorupee.py"
Enter Dollar Amount : 100
Entered Dollar Amount :  100.0
$ :  100.0
₹ :  8628.0
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

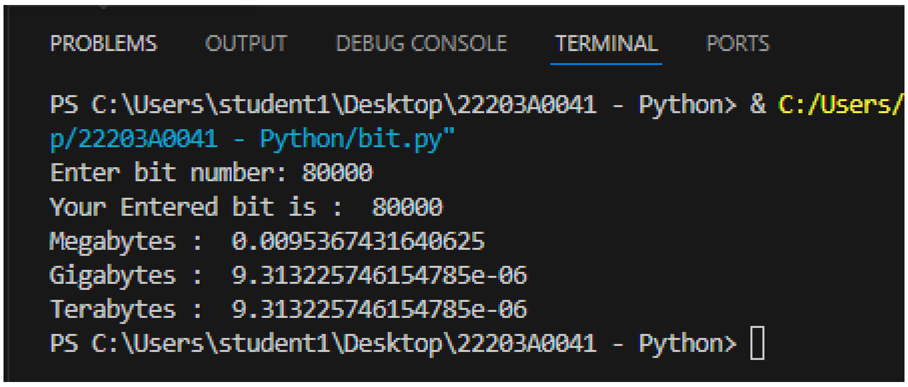
2. Write a program to convert bits to Megabytes, Gigabytes and Terabytes.

Ans:

a] Code:

```
bit = int(input("Enter bit number: "))

print("Your Entered bit is : " , bit)

byte = bit/8

kb = byte/1024

mb = kb/1024

gb = mb/1024

tb = mb/1024

print("Megabytes : " , mb )

print("Gigabytes : " , gb )

print("Terabytes : " , tb )
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/Users/
p/22203A0041 - Python/bit.py"
Enter bit number: 80000
Your Entered bit is :  80000
Megabytes :  0.0095367431640625
Gigabytes :  9.313225746154785e-06
Terabytes :  9.313225746154785e-06
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

3. Write a program to find the square root of a number.

Ans:

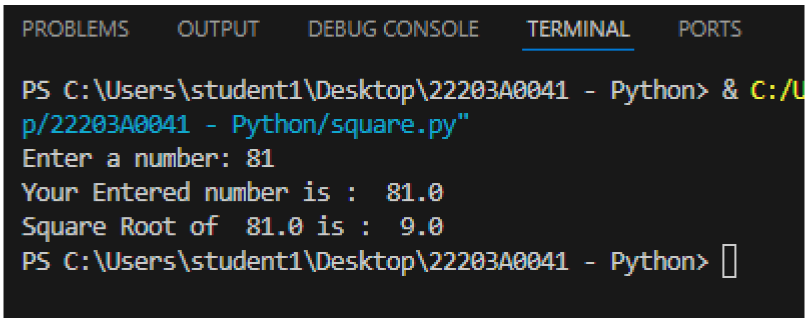
a] Code:

num = float(input("Enter a number: "))

print("Your Entered number is : " , num)

print("Square Root of " , num , "is : " , (num ** 0.5))

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/U
p/22203A0041 - Python/square.py"
Enter a number: 81
Your Entered number is :  81.0
Square Root of  81.0 is :  9.0
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

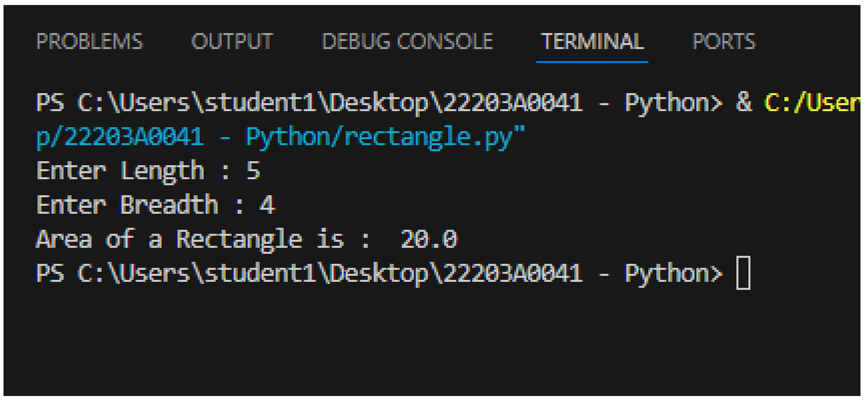
4. Write a program to find the area of Rectangle.

Ans:

a] Code:

```
len = float(input("Enter Length : " ))

bre = float(input("Enter Breadth : " ))

aor = len * bre

print("Area of a Rectangle is : " , aor )
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/User
p/22203A0041 - Python/rectangle.py"
Enter Length : 5
Enter Breadth : 4
Area of a Rectangle is :  20.0
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

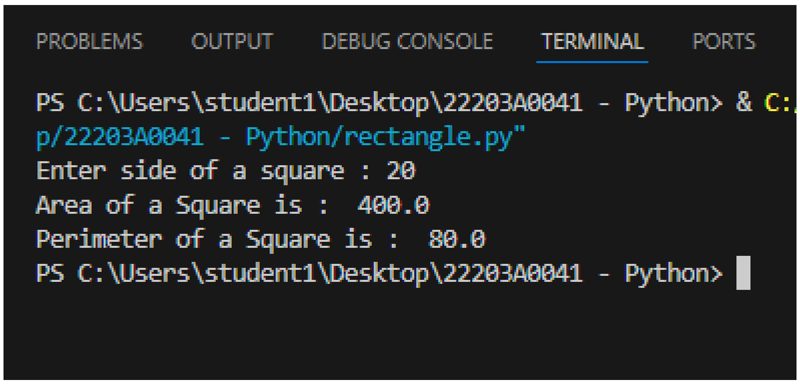
5. Write a program to calculate area and perimeter of the square

Ans:

a] Code:

```
side = float(input("Enter side of a square : "))

area = side * side

per = 4 * side

print("Area of a Square is : " , area)

print("Perimeter of a Square is : " , per)
```

b] Output:


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:
p/22203A0041 - Python/rectangle.py"
Enter side of a square : 20
Area of a Square is :  400.0
Perimeter of a Square is :  80.0
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

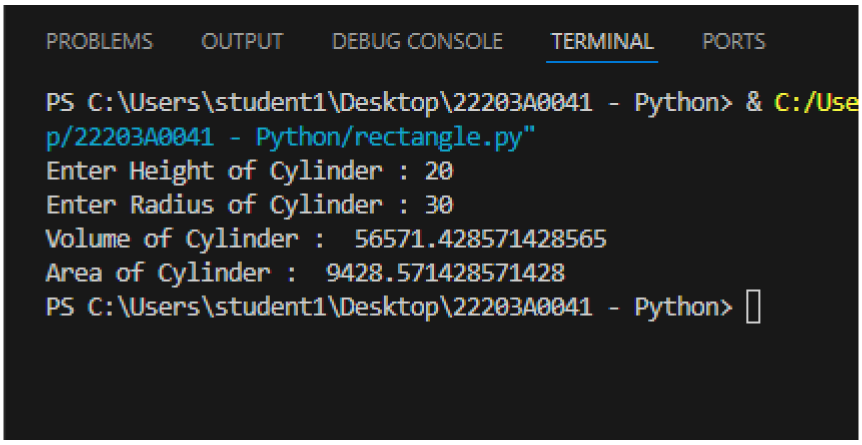
6. Write a program to calculate surface volume and area of a cylinder.

Ans:

<u>a] Code:</u>

```
pi = 22/7

height = float(input("Enter Height of Cylinder : "))

radius = float(input("Enter Radius of Cylinder : "))

volume = pi * radius * radius * height

area = (2 * pi * radius * height) + (2 * pi * radius ** 2)

print("Volume of Cylinder : " , volume)

print("Area of Cylinder : " , area)
```

<u>b] Output:</u>


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/Use
p/22203A0041 - Python/rectangle.py"
Enter Height of Cylinder : 20
Enter Radius of Cylinder : 30
Volume of Cylinder :  56571.428571428565
Area of Cylinder :  9428.571428571428
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

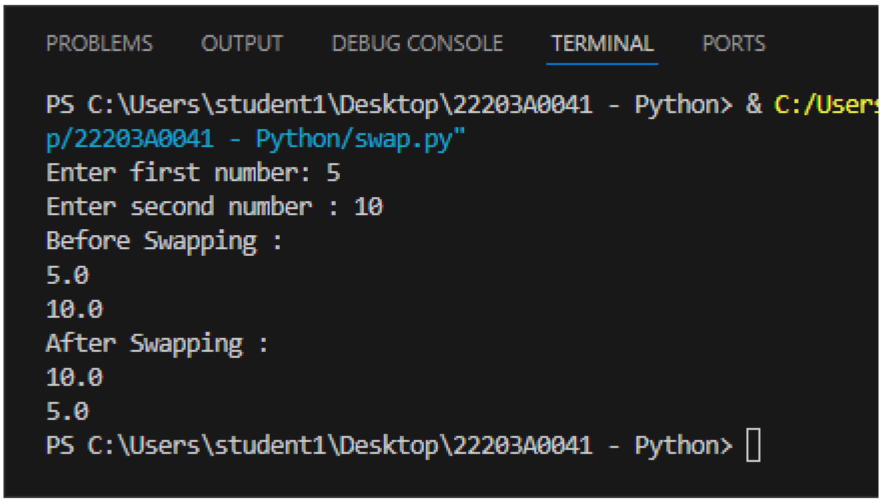
7. Write a program to swap the value of two variables.

Ans:

a] Code:

```
x = float(input("Enter first number: "))

y = float(input("Enter second number : "))

print("Before Swapping : ")

print(x)

print(y)

temp = x

x = y

y = temp

print("After Swapping : ")

print(x)

print(y)
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/Users
p/22203A0041 - Python/swap.py"
Enter first number: 5
Enter second number : 10
Before Swapping :
5.0
10.0
After Swapping :
10.0
5.0
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	4
Title of Experiment	Write simple Python program to demonstrate use of conditional statements: if' statement, 'if … else' statement, Nested 'if' statement

X. PRACTICAL RELATED QUESTIONS

1. List operators used in if conditional statement ?

ANS:

1.Conditional operators used in if conditional statement :-

- **==**: Equal to
- **!=**: Not equal to
- **>**: Greater than
- **<**: Less than
- **>=**: Greater than or equal to
- **<=**: Less than or equal to

2. Differentiate between if-else and nested-if statement ?

Ans:

Feature	**if-else**	**Nested-if**
Structure	One `if` block and one `else` block	An `if` block inside another `if` block (or `else` block)
Complexity	Simpler, handles a single condition	More complex, handles multiple conditions in layers
Use case	Used when you have two choices (if or else)	Used when you have multiple conditions to check in sequence
Flow	Executes either `if` block or `else` block	Executes blocks based on nested conditions
Example	`if age >= 18: ... else: ...`	`if age >= 18: if income >= 40000: ... else: ...`

XI. Exercise

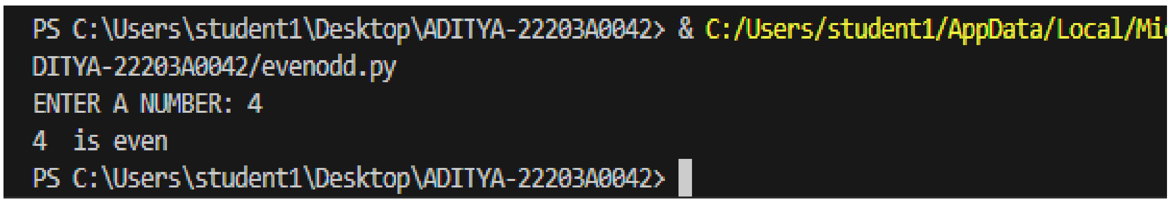
1. Write a program to check whether a number is even or odd ?

Ans:

a] Code:

```
n=int(input("ENTER A NUMBER: "))


if n%2==0:
    print(n," is even")
else:
    print(n," is odd")
```

b] Output:



```
PS C:\Users\student1\Desktop\ADITYA-22203A0042> & C:/Users/student1/AppData/Local/Mi
DITYA-22203A0042/evenodd.py
ENTER A NUMBER: 4
4  is even
PS C:\Users\student1\Desktop\ADITYA-22203A0042> 
```

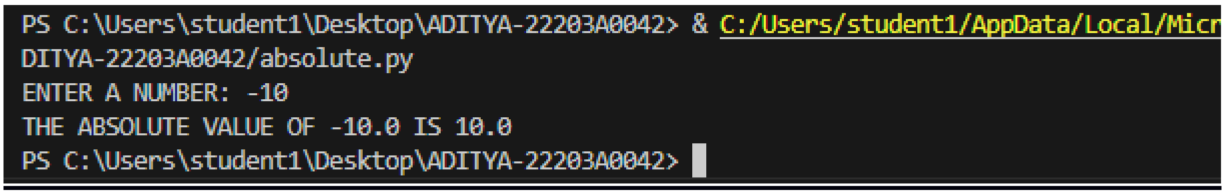
2. Write a program to find out absolute value of an input number ?

Ans:

a] Code:

```
n=float(input("ENTER A NUMBER: "))
print("THE ABSOLUTE VALUE OF",n,"IS",abs(n))
```

b] Output:



```
PS C:\Users\student1\Desktop\ADITYA-22203A0042> & C:/Users/student1/AppData/Local/Micr
DITYA-22203A0042/absolute.py
ENTER A NUMBER: -10
THE ABSOLUTE VALUE OF -10.0 IS 10.0
PS C:\Users\student1\Desktop\ADITYA-22203A0042> 
```

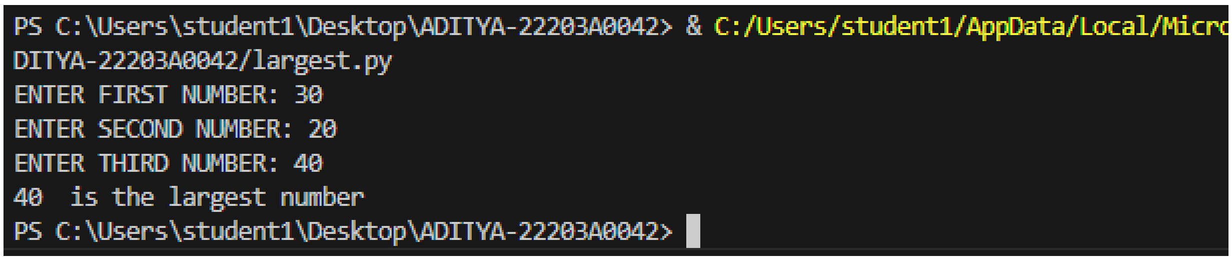
3. Write a program to check the largest number among the three numbers ?

Ans:

a] Code:

```
n1=int(input("ENTER FIRST NUMBER: "))

n2=int(input("ENTER SECOND NUMBER: "))

n3=int(input("ENTER THIRD NUMBER: "))


if n1>n2:

   if n1>n3:

      print(n1," is the largest number")

   else:

      print(n3," is the largest number")

else:

   if n2>n3:

      print(n2," is the largest number")

   else:

      print(n3," is the largest number")
```

b] Output:



```
PS C:\Users\student1\Desktop\ADITYA-22203A0042> & C:/Users/student1/AppData/Local/Micro
DITYA-22203A0042/largest.py
ENTER FIRST NUMBER: 30
ENTER SECOND NUMBER: 20
ENTER THIRD NUMBER: 40
40  is the largest number
PS C:\Users\student1\Desktop\ADITYA-22203A0042>
```

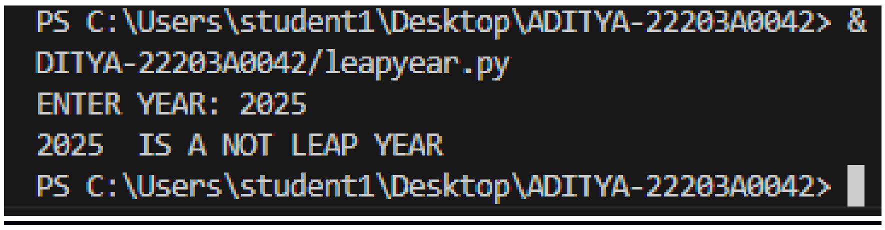
4. Write a program to check if the input year is a leap year of not ?

Ans:

<u>a] Code:</u>

```
year=int(input("ENTER YEAR: "))


if year%4==0:
    print(year," IS A LEAP YEAR")
else:
    print(year," IS A NOT LEAP YEAR")
```

<u>b] Output:</u>



```
PS C:\Users\student1\Desktop\ADITYA-22203A0042> &
DITYA-22203A0042/leapyear.py
ENTER YEAR: 2025
2025  IS A NOT LEAP YEAR
PS C:\Users\student1\Desktop\ADITYA-22203A0042>
```

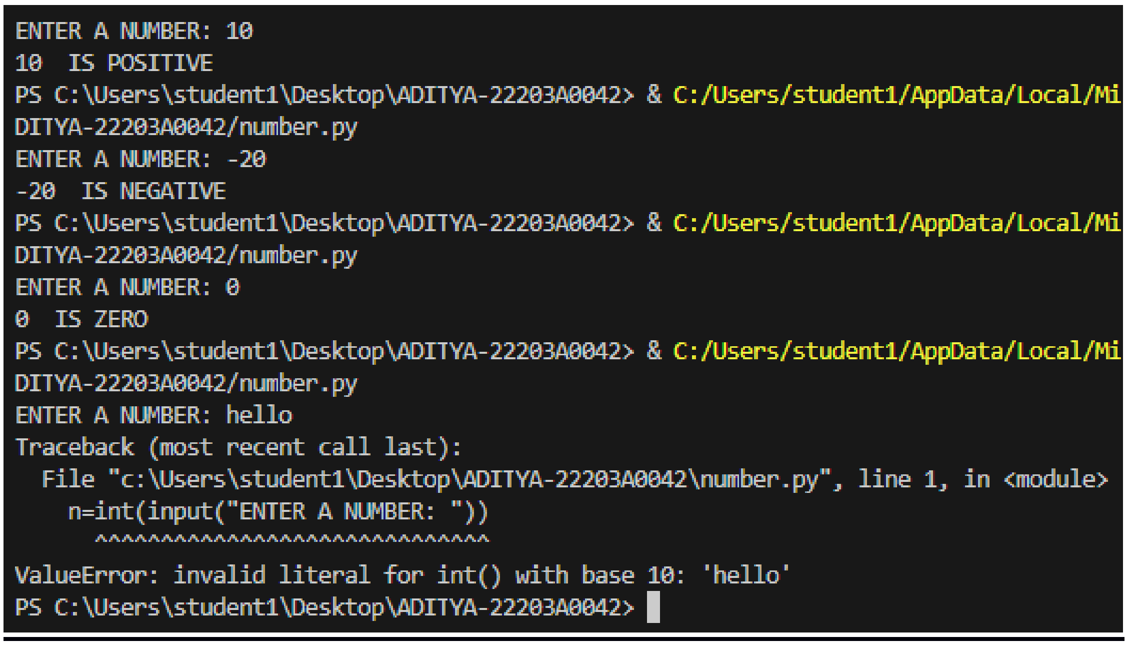
5. Write a program to check if a Number is Positive, Negative or Zero ?

Ans:

<u>a] Code:</u>

```
n=int(input("ENTER A NUMBER: "))


if n<0:
    print(n," IS NEGATIVE")
elif n>0:
    print(n," IS POSITIVE")
elif n==0:
    print(n," IS ZERO")

else:

    print("Invalid input")
```

b] Output:



```
ENTER A NUMBER: 10
10  IS POSITIVE
PS C:\Users\student1\Desktop\ADITYA-22203A0042> & C:/Users/student1/AppData/Local/Mi
DITYA-22203A0042/number.py
ENTER A NUMBER: -20
-20  IS NEGATIVE
PS C:\Users\student1\Desktop\ADITYA-22203A0042> & C:/Users/student1/AppData/Local/Mi
DITYA-22203A0042/number.py
ENTER A NUMBER: 0
0  IS ZERO
PS C:\Users\student1\Desktop\ADITYA-22203A0042> & C:/Users/student1/AppData/Local/Mi
DITYA-22203A0042/number.py
ENTER A NUMBER: hello
Traceback (most recent call last):
  File "c:\Users\student1\Desktop\ADITYA-22203A0042\number.py", line 1, in <module>
    n=int(input("ENTER A NUMBER: "))
      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 10: 'hello'
PS C:\Users\student1\Desktop\ADITYA-22203A0042>
```

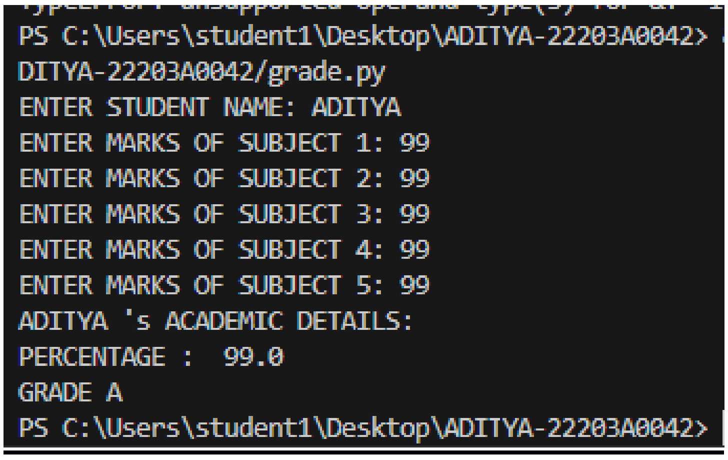
6. Write a program that takes the marks of 5 subjects and displays the grade ?

Ans:

a] Code:

```
name=input("ENTER STUDENT NAME: ")

m1=float(input("ENTER MARKS OF SUBJECT 1: "))

m2=float(input("ENTER MARKS OF SUBJECT 2: "))

m3=float(input("ENTER MARKS OF SUBJECT 3: "))

m4=float(input("ENTER MARKS OF SUBJECT 4: "))

m5=float(input("ENTER MARKS OF SUBJECT 5: "))



per=((m1+m2+m3+m4+m5)/500)*100



print(name,"'s ACADEMIC DETAILS: ")
print("PERCENTAGE : ",per)


if per>=90 and per<=100:
   print("GRADE A")
elif per>=75 and per<90:
   print("GRADE B")
elif per>=45 and per<75:
   print("GRADE B")
elif per>=0 and per<45:
   print("fail !")
else:
   print("INVALID INPUT !")
```

b] Output:



```
PS C:\Users\student1\Desktop\ADITYA-22203A0042>
DITYA-22203A0042/grade.py
ENTER STUDENT NAME: ADITYA
ENTER MARKS OF SUBJECT 1: 99
ENTER MARKS OF SUBJECT 2: 99
ENTER MARKS OF SUBJECT 3: 99
ENTER MARKS OF SUBJECT 4: 99
ENTER MARKS OF SUBJECT 5: 99
ADITYA 's ACADEMIC DETAILS:
PERCENTAGE :  99.0
GRADE A
PS C:\Users\student1\Desktop\ADITYA-22203A0042>
```

Vidyalankar Polytechnic	DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming With Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	5
Title of Experiment	Write Python program to demonstrate use of looping statements: 'while' loop, 'for' loop and Nested loop

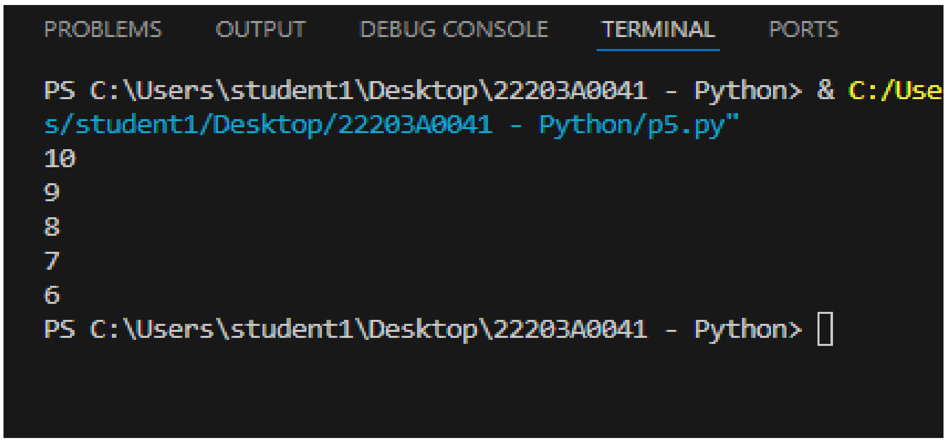
X. PRACTICAL RELATED QUESTIONS

1. What would be the output from the following Python code segment?

```
x=10
while x>5:
print x,
x-=1
```

ANS:

a] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/Use
s/student1/Desktop/22203A0041 - Python/p5.py"
10
9
8
7
6
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

2. Change the following Python code from using a while loop to for loop:

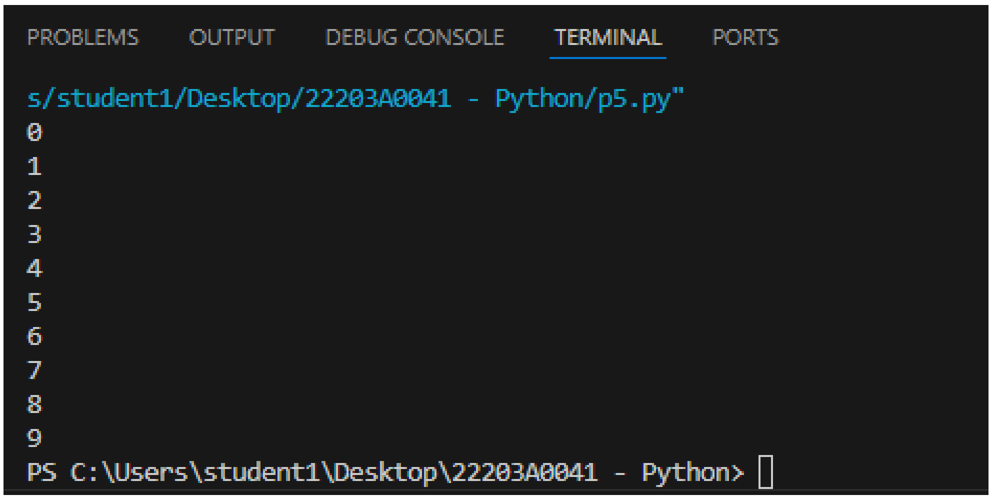
```
x=1
while x
print x,
x+=1
```

ANS:

a] Code:

```
for i in range(10):
   print(i)
```

b] Output:



```
s/student1/Desktop/22203A0041 - Python/p5.py"
0
1
2
3
4
5
6
7
8
9
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

XI Exercise

1. Print the following patterns using loop:

ANS:

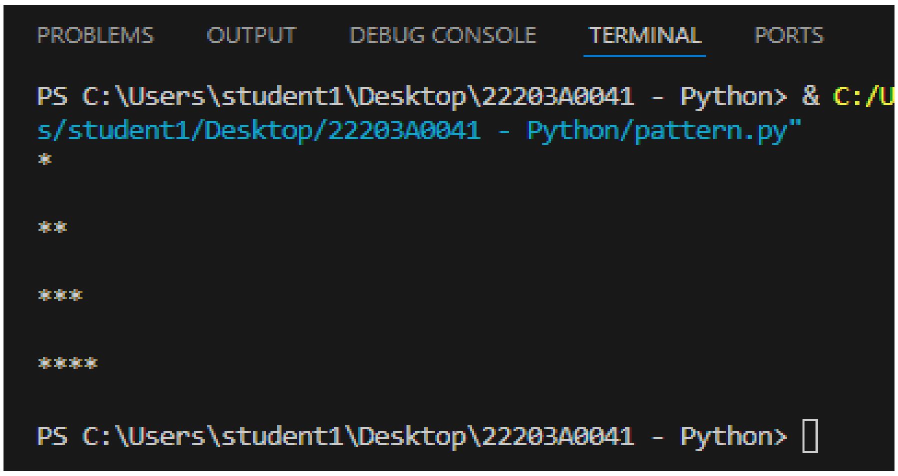
a]

```
*

**

***

****
```

Code:

```
for i in range(0,4):
    for j in range(0,i+1):
        print("*" ,end="")
    print("\n")
```

Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/U
s/student1/Desktop/22203A0041 - Python/pattern.py"
*

**

***

****

PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

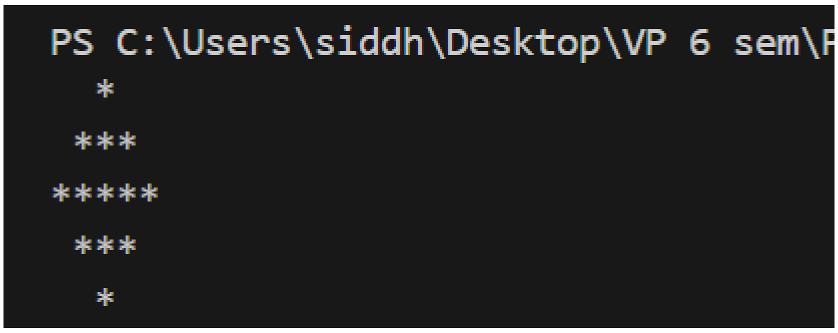
b]

```
  *
 ***
*****
 ***
  *
```

Code:

```
p = [1, 3, 5, 3, 1]
for i in range(5):
    print(" " * abs(2 - i) + "*" * p[i])
```

Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\F
  *
 ***
*****
 ***
  *
```

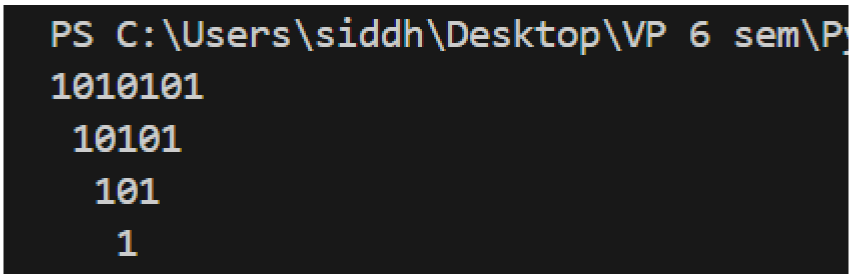
c]

```
1010101
 10101
  101
   1
```

Code:

```
n = 7
for i in range(0, n, 2):
    print(" " * (i // 2) + "10" * ((n - i) // 2) + "1" + " " * (i // 2))
```

Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\P
1010101
 10101
  101
   1
```

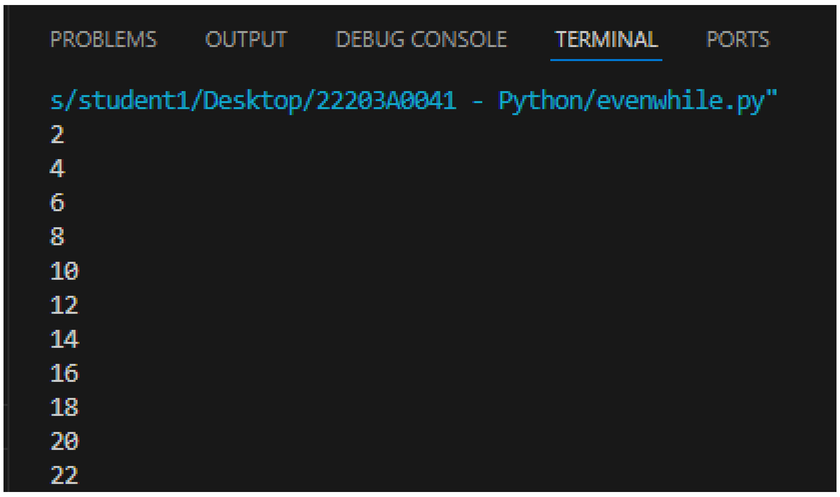
2. Write a Python program to print all even numbers between 1 to 100 using while loop.

Ans:

a] Code:

```
i = 0
while i < 100:
    i += 2
    print(i)
else:
    print("End of While Loop")
```

b] Output:



```
s/student1/Desktop/22203A0041 - Python/evenwhile.py"
2
4
6
8
10
12
14
16
18
20
22
```

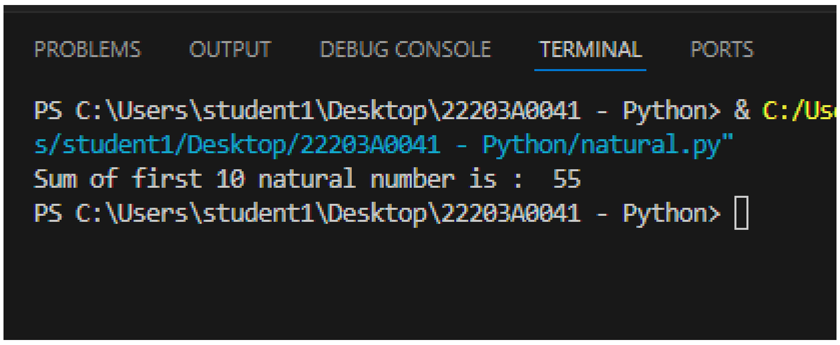
3. Write a Python program to find the sum of first 10 natural numbers using for loop

ANS:

a] Code:

```
sum = 0
for i in range(11):
    sum += i
print("Sum of first 10 natural number is : ",sum)
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/Us
s/student1/Desktop/22203A0041 - Python/natural.py"
Sum of first 10 natural number is :  55
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

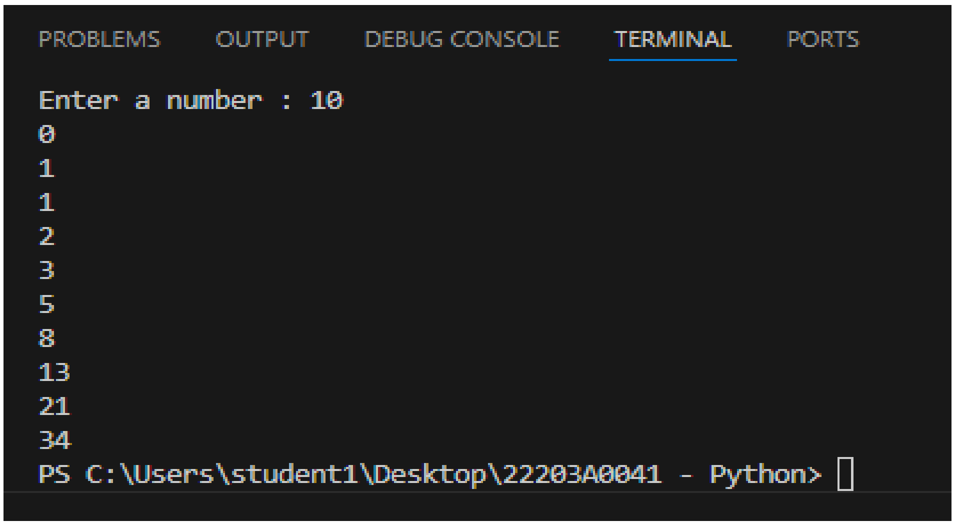
4. Write a Python program to print Fibonacci series.

ANS:

a] Code:

```
n1 = 0
n2 = 1
n3 = 0
num = int(input("Enter a number : "))
for i in range(num):
    print(n1)
    n3 = n1 + n2
    n1 = n2
    n2 = n3
```

b] Output:



```
Enter a number : 10
0
1
1
2
3
5
8
13
21
34
PS C:\Users\student1\Desktop\22203A0041 - Python>
```

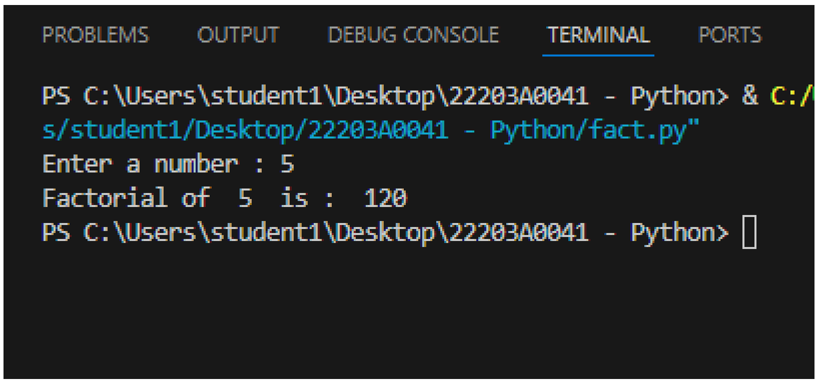
5. Write a Python program to calculate factorial of a number

ANS:

a] Code:

```
num = int(input("Enter a number : "))

fact = 1

for i in range(1,num+1):

    fact *= i

print("Factorial of ",num, " is : ",fact)
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/
s/student1/Desktop/22203A0041 - Python/fact.py"
Enter a number : 5
Factorial of  5  is :  120
PS C:\Users\student1\Desktop\22203A0041 - Python>
```

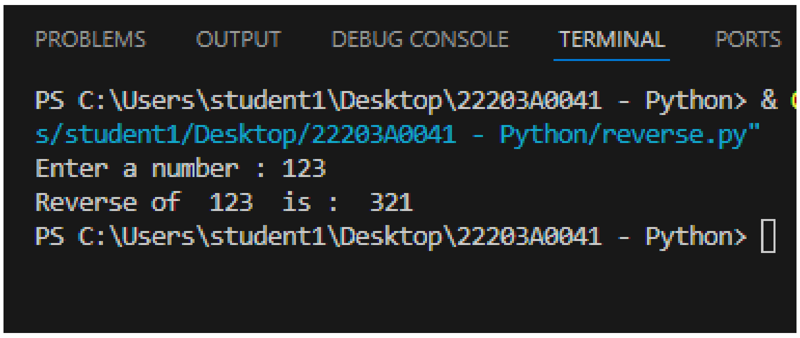
6. Write a Python Program to Reverse a Given Number

ANS:

a] Code:

```
num = int(input("Enter a number : "))

rev = 0

d = 0

n = num

while n!=0:

    d = n%10

    rev = (rev*10)+d

    n = n//10

else:

    print("Reverse of ",num, " is : ",rev)
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & 
s/student1/Desktop/22203A0041 - Python/reverse.py"
Enter a number : 123
Reverse of  123  is :  321
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

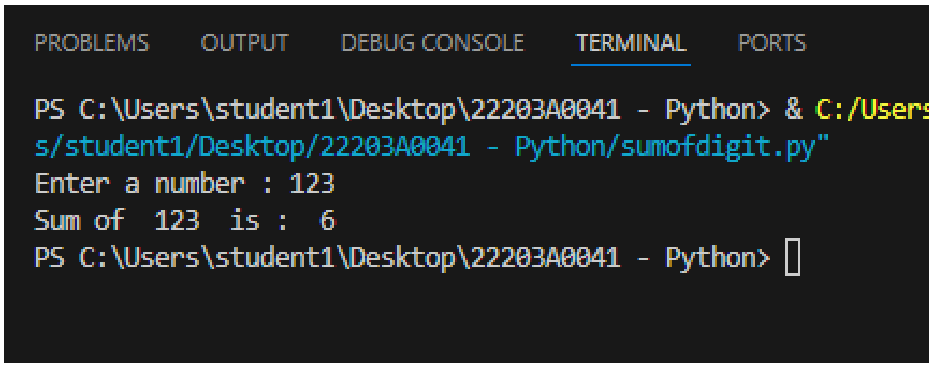
7. Write a Python program takes in a number and finds the sum of digits in a number.

ANS:

a] Code:

```
num = int(input("Enter a number : "))

n = num

total = 0

while n!=0:
    total += n%10
    n //=10
print("Sum of ",num, " is : ",total)
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/Users
s/student1/Desktop/22203A0041 - Python/sumofdigit.py"
Enter a number : 123
Sum of  123  is :  6
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

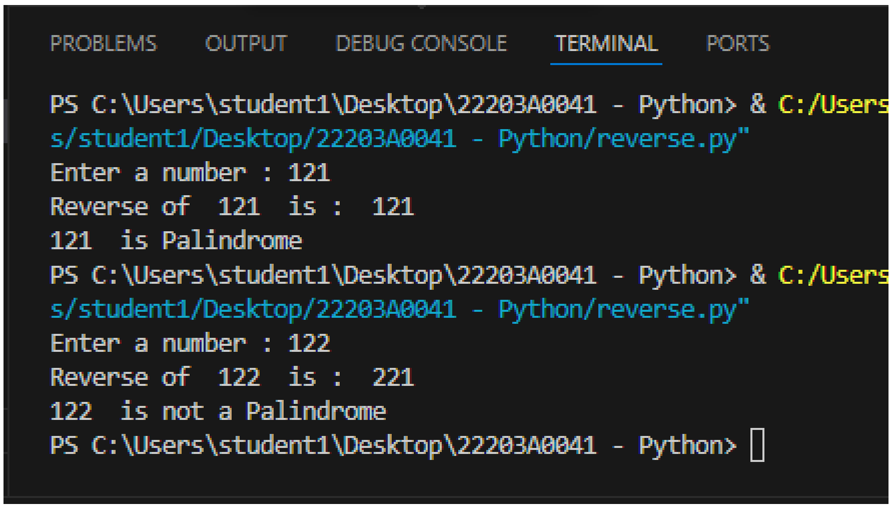
8. Write a Python program that takes a number and checks whether it is a palindrome or not.

ANS:

a] Code:

```
num = int(input("Enter a number : "))
rev = 0
d = 0
n = num
while n!=0:
    d = n%10
    rev = (rev*10)+d
    n = n//10
else:
    print("Reverse of ",num, " is : ",rev)
    if(rev == num):
        print(num, " is Palindrome")
    else:
        print(num, " is not a Palindrome")
```

b] Output:



```
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/Users
s/student1/Desktop/22203A0041 - Python/reverse.py"
Enter a number : 121
Reverse of  121  is :  121
121  is Palindrome
PS C:\Users\student1\Desktop\22203A0041 - Python> & C:/Users
s/student1/Desktop/22203A0041 - Python/reverse.py"
Enter a number : 122
Reverse of  122  is :  221
122  is not a Palindrome
PS C:\Users\student1\Desktop\22203A0041 - Python> 
```

Vidyalankar Polytechnic

DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	6
Title of Experiment	Write Python program to perform following operations on Lists: Create list, Access list, Update list (Add item, Remove item), Delete list

X. PRACTICAL RELATED QUESTIONS

1. When to used list ?

ANS:

In Python, lists are used when you need to store a collection of items .You typically use lists in these scenarios:

- When you need to store a sequence of items: For example, if you're managing a list of numbers, strings, or objects .
- When order matters: If you need to keep items in a specific order and might need to update them later.
- When you might need to change or modify the collection: Lists are good when you plan to add or remove items over time.

2. Describe various list functions ?

Ans:

1. append()

- Adds an item to the end of the list.
- Syntax: list.append(item)

2. extend()

- Adds all elements from an iterable (like another list, tuple, or string) to the end of the list.
- Syntax: list.extend(iterable)

3. insert()

- Inserts an item at a specified index in the list.
- Syntax: list.insert(index, item)

4. remove()

- Removes the first occurrence of a specified item. If the item is not found, it raises a ValueError.
- Syntax: list.remove(item)

5. pop()

- Removes and returns an item at a specified index (default is the last item). If the list is empty, it raises an IndexError.
- Syntax: list.pop([index])

6. clear()

- Removes all items from the list, leaving it empty.
- Syntax: list.clear()

7. index()

- Returns the index of the first occurrence of the specified item. If the item is not found, it raises a ValueError.
- Syntax: list.index(item)

8. count()

- Returns the number of occurrences of a specified item in the list.
- Syntax: list.count(item)

9. sort()

- Sorts the items of the list in place (modifies the list).

○ You can specify the sorting order (ascending or descending) with the reverse parameter.

- Syntax: list.sort(reverse=False)
- To sort in reverse (descending) order:

fruits.sort(reverse=True)

10. copy()

- Returns a copy of the list.
- Syntax: list.copy()

11. join()

- Converts a list of strings into a single string, with a specified separator between each element.
- Syntax: separator.join(list)

12. list()

- Creates a list from an iterable (such as a string, tuple, or set).
- Syntax: list(iterable)

13. max() and min()

- Finds the maximum and minimum values in a list.
- Syntax: max(list) or min(list)

3. Write syntax for a method to sort a list ?

Ans:

- Sorts the items of the list in place (modifies the list).
- You can specify the sorting order (ascending or descending) with the reverse parameter.
- Syntax: list.sort(reverse=False)

Example

```
fruits = ["apple", "cherry", "banana"]

fruits.sort()

print(fruits)  # Output: ['apple', 'banana', 'cherry']
```

Example

To sort in reverse (descending) order:-

```
fruits.sort(reverse=True)

print(fruits)  # Output: ['cherry', 'banana', 'apple']
```

4. Write syntax for a method to count occurrences of a list item in Python ?

Ans:

To count the occurrences of an item in a list in Python, you can use the count() method. Here's the syntax:

```
list.count(item)
```

- list: The list in which you want to count the occurrences.
- item: The element whose occurrences you want to count.

Example:

```
fruits = ["apple", "banana", "cherry", "banana", "apple", "banana"]

count_banana = fruits.count("banana")

print(count_banana)  # Output: 3
```

5. How to concatenate list ?

Ans:

In Python, we can concatenate lists using a two different methods:

1. Using the + Operator

The simplest way to concatenate two or more lists is by using the + operator. This creates a new list that combines the elements of both lists.

Syntax:

```
list1 + list2
```

Example:

```
list1 = [1, 2, 3]

list2 = [4, 5, 6]

concatenated_list = list1 + list2

print(concatenated_list)  # Output: [1, 2, 3, 4, 5, 6]
```

2. Using extend() Method

The extend() method adds all elements from one list to the end of another list, modifying the original list in place.

Syntax:

```
list1.extend(list2)
```

Example:

```
list1 = [1, 2, 3]

list2 = [4, 5, 6]

list1.extend(list2)

print(list1)  # Output: [1, 2, 3, 4, 5, 6]
```

6. Justify the statement "Lists are mutable" ?

Ans:

In Python, lists are considered **mutable** because their elements can be changed after the list is created. You can modify individual elements by using their indices, add new elements using methods like append() or insert(), and remove items with methods such as remove() or pop(). Additionally, the size of a list can be altered by adding or removing elements, and operations like sort() modify the list in place without creating a new one. This ability to change, add, and remove elements demonstrates that lists are mutable objects in Python.

7. Describe the use pop operator in list ?

Ans:

In Python, the pop() method removes and returns an item from a list. By default, it removes the last item, but you can also specify an index to remove an item at a specific position.

Syntax:

```
list.pop([index])
```

- If no index is provided, it removes and returns the last item.
- If an index is provided, it removes and returns the item at that index.

Examples:

1. Removing the last item:

```
fruits = ["apple", "banana", "cherry"]

removed_item = fruits.pop()

print(removed_item)  # Output: 'cherry'
```

2. Removing an item at a specific index:

```
fruits = ["apple", "banana", "cherry"]

removed_item = fruits.pop(1)

print(removed_item)  # Output: 'banana'
```

If the list is empty, pop() raises an IndexError.

Use Case:

pop() is useful when you need to remove and get an item from a list at the same time.

XI. Exercise

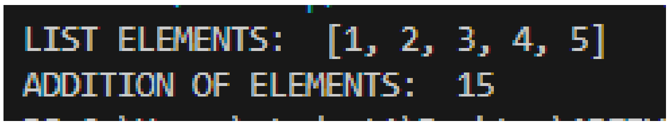
1. **Write a Python program to sum all the items in a list ?**

Ans:

a] Code1:

```
l1=[1,2,3,4,5]


print("LIST ELEMENTS: ",l1)

print("ADDITION OF ELEMENTS: ", sum(l1))
```

b] Output:



```
LIST ELEMENTS:  [1, 2, 3, 4, 5]
ADDITION OF ELEMENTS:  15
```

2. Write a Python program to multiplies all the items in a list ?

Ans:

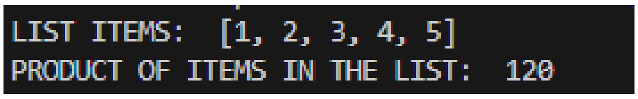
a] Code:

```
l1=[1,2,3,4,5]

mul=1

print("LIST ITEMS: ",l1)

for i in l1:

   mul*=i
```

else:

    print("PRODUCT OF ITEMS IN THE LIST: ",mul)

b] Output:



```
LIST ITEMS:  [1, 2, 3, 4, 5]
PRODUCT OF ITEMS IN THE LIST:  120
```

3. Write a Python program to get the largest number from a list ?

Ans:

a] Code:

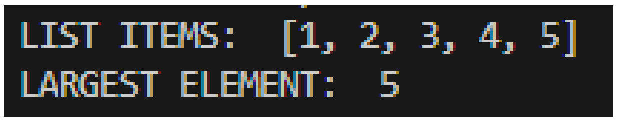
l1=[1,2,3,4,5]

mul=1

print("LIST ITEMS: ",l1)

print("LARGEST ELEMENT: ",max(l1))

b] Output:



```
LIST ITEMS:  [1, 2, 3, 4, 5]
LARGEST ELEMENT:  5
```

4. Write a Python program to get the smallest number from a list ?

Ans:

a] Code:

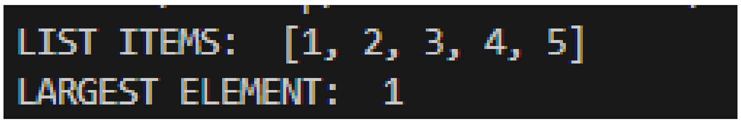
l1=[1,2,3,4,5]

mul=1

print("LIST ITEMS: ",l1)

print("LARGEST ELEMENT: ",min(l1))

b] Output:



```
LIST ITEMS:  [1, 2, 3, 4, 5]
LARGEST ELEMENT:  1
```

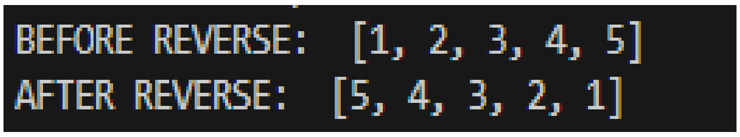
5. Write a Python program to reverse a list ?

Ans:

a] Code:

```
l1=[1,2,3,4,5]


print("BEFORE REVERSE: ", l1)


l1.reverse()

print("AFTER REVERSE: ", l1)
```

b] Output:



```
BEFORE REVERSE:  [1, 2, 3, 4, 5]
AFTER REVERSE:  [5, 4, 3, 2, 1]
```

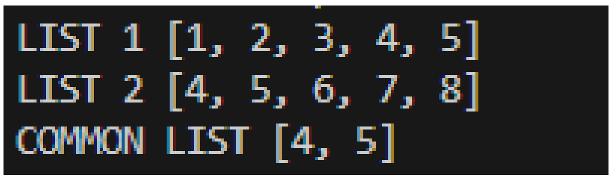
6. Write a Python program to find common items from two lists ?

Ans:

a] Code:

```
l1=[1,2,3,4,5]

l2=[4,5,6,7,8]

l3=list(set(l1) & set(l2))

print("LIST 1",l1)

print("LIST 2",l2)

print("COMMON LIST",l3)
```

<u>b] Output:</u>



```
LIST 1 [1, 2, 3, 4, 5]
LIST 2 [4, 5, 6, 7, 8]
COMMON LIST [4, 5]
```

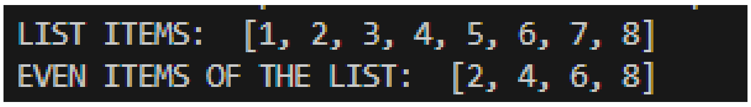
7. Write a Python program to select the even items of a list ?

Ans:

<u>a] Code:</u>

```
l1=[1,2,3,4,5,6,7,8]

even=[]


print("LIST ITEMS: ",l1)

for i in l1:

    if i%2==0:

        even.append(i)

else:

    print("EVEN ITEMS OF THE LIST: ",even)
```

<u>b] Output:</u>



```
LIST ITEMS:  [1, 2, 3, 4, 5, 6, 7, 8]
EVEN ITEMS OF THE LIST:  [2, 4, 6, 8]
```

Vidyalankar Polytechnic	DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	7
Title of Experiment	Write Python program to perform following operations on Tuples: Create Tuple, Access Tuple, Update Tuple, Delete Tuple

X. PRACTICAL RELATED QUESTIONS

1. Define empty tuple. Write syntax to create empty tuple ?

ANS:

An empty tuple in Python is a tuple that contains no elements. It is created using a pair of empty parentheses ().

Syntax to create an empty tuple:

```
empty_tuple = ()
```

Example:

```
empty_tuple = ()
print(type(empty_tuple))  # Output: <class 'tuple'>
```

2. Write syntax to copy specific elements existing tuple into new tuple ?

Ans:

To copy specific elements from an existing tuple into a new tuple, we can use **slicing**. This allows you to select

a portion of the original tuple and create a new one.

Syntax:

```
new_tuple = old_tuple[start:end]
```

- start: The index where you want to begin copying (inclusive).

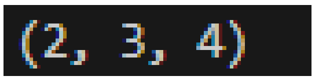
- end: The index where you want to stop copying (exclusive).

Example:

```
old_tuple = (1, 2, 3, 4, 5)

new_tuple = old_tuple[1:4]


print(new_tuple)  # Output: (2, 3, 4)
```

In this example, we used slicing to copy the elements from index 1 to 3 (not including index

This creates a new tuple new_tuple with the values (2, 3, 4).



```
(2, 3, 4)
```

3. Write syntax for a method to sort a list ?

Ans:

Feature	Tuple	List
Mutability	Immutable (cannot be changed after creation)	Mutable (can be modified after creation)
Syntax	Defined using parentheses ()	Defined using square brackets []
Performance	Faster than lists, especially for iteration and access	Slower than tuples due to mutability overhead
Use Case	Best suited for fixed data or data that shouldn't change	Best suited for collections of items that may need to be changed

XI. Exercise

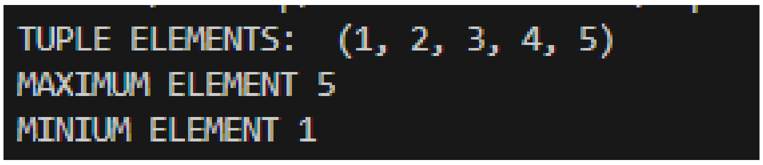
1. **Create a tuple and find the minimum and maximum number from it ?**

Ans:

a] Code1:

```
t1=(1,2,3,4,5)


print("TUPLE ELEMENTS: ", t1)

print("MAXIMUM ELEMENT",max(t1))

print("MINIUM ELEMENT",min(t1))
```

b] Output:



```
TUPLE ELEMENTS:  (1, 2, 3, 4, 5)
MAXIMUM ELEMENT 5
MINIUM ELEMENT 1
```

2. Write a Python program to find the repeated items of a tuple ?

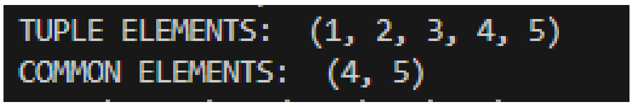
Ans:

a] Code:

```
t1=(1,2,3,4,5)

t2=(4,5,6,7,8)

print("TUPLE ELEMENTS: ", t1)

common=[]

for i in range(len(t1)):

    if t1[i] in t2:

        common.append(t1[i])
```

```
else:
    com=tuple(common)
    print("COMMON ELEMENTS: ", com)
```

<u>b] Output:</u>



```
TUPLE ELEMENTS:  (1, 2, 3, 4, 5)
COMMON ELEMENTS:  (4, 5)
```

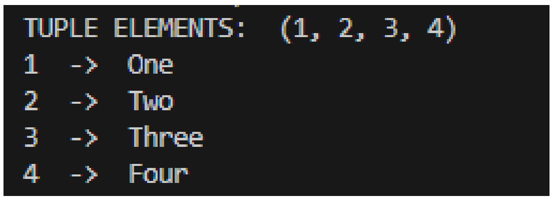
3. Print the number in words for Example: 1234 => One Two Three Four ?

Ans:

<u>a] Code:</u>

```
t1=(1,2,3,4)
t2=("One","Two","Three","Four")


print("TUPLE ELEMENTS: ", t1)
common=[]
for i in range(len(t1)):
    print(t1[i]," -> ",t2[i])
```

<u>b] Output:</u>



```
TUPLE ELEMENTS:  (1, 2, 3, 4)
1  ->  One
2  ->  Two
3  ->  Three
4  ->  Four
```

	DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	8
Title of Experiment	Write Python program to perform following operations on Set: Create Set, Access Set elements, Update Set, Delete Set

PRACTICAL RELATED QUESTIONS:

1. Describe the various set operations

Ans:

Intersection

The intersection of the sets A and B, denoted by A ∩ B, is the set of elements that belong to both A and B i.e. set of the common elements in A and B.

Disjoint Set

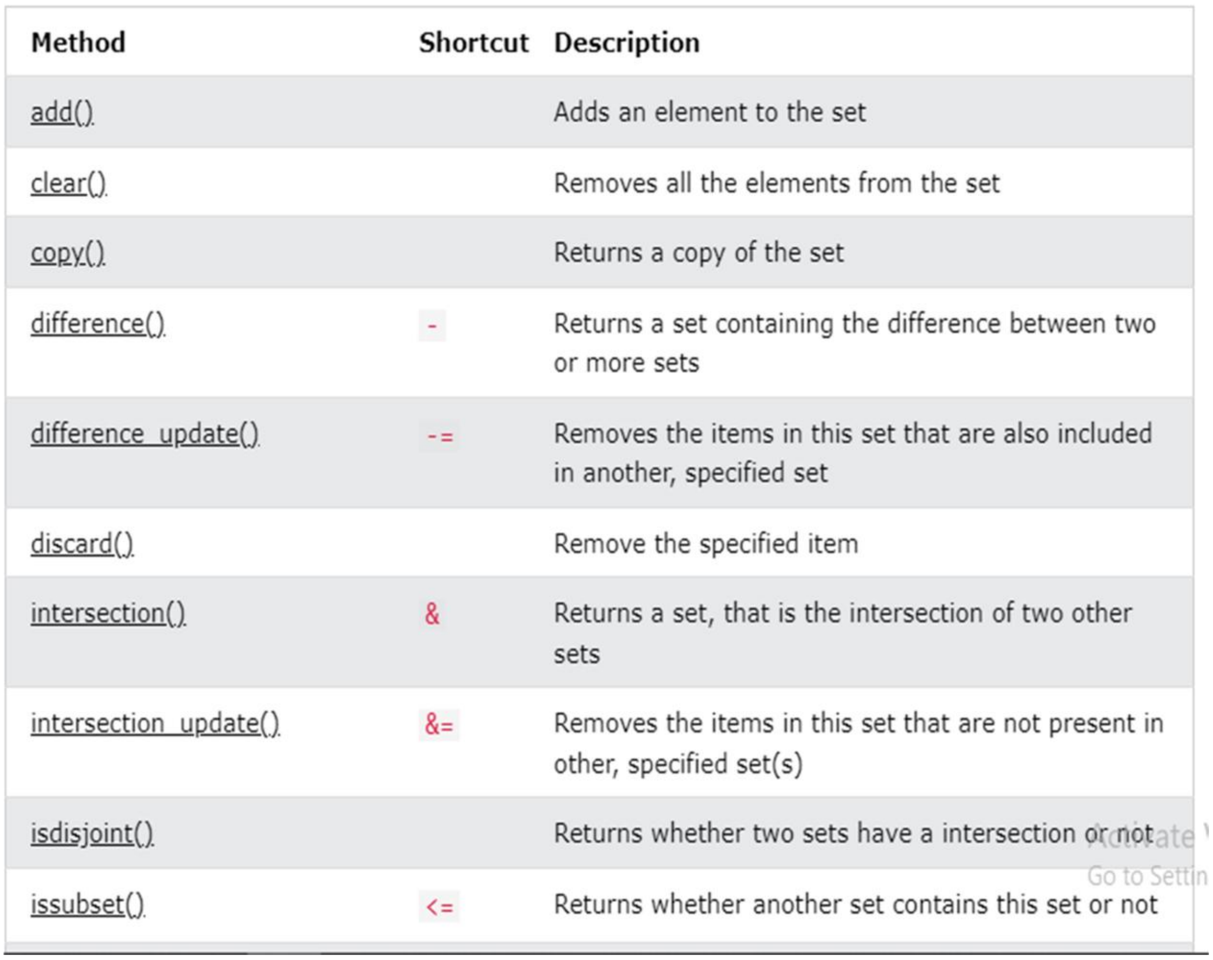
Two sets are said to be disjoint if their intersection is the empty set. i.e, sets have no common elements.

Union of Sets

Union of sets A and B, denoted by A ∪ B, is the set of distinct elements that belong to set A or set B, or both.

2. Describe the various methods of set

Ans:



Method	Shortcut	Description
add()		Adds an element to the set
clear()		Removes all the elements from the set
copy()		Returns a copy of the set
difference()	-	Returns a set containing the difference between two or more sets
difference_update()	-=	Removes the items in this set that are also included in another, specified set
discard()		Remove the specified item
intersection()	&	Returns a set, that is the intersection of two other sets
intersection_update()	&=	Removes the items in this set that are not present in other, specified set(s)
isdisjoint()		Returns whether two sets have a intersection or not
issubset()	<=	Returns whether another set contains this set or not

EXERCISE:

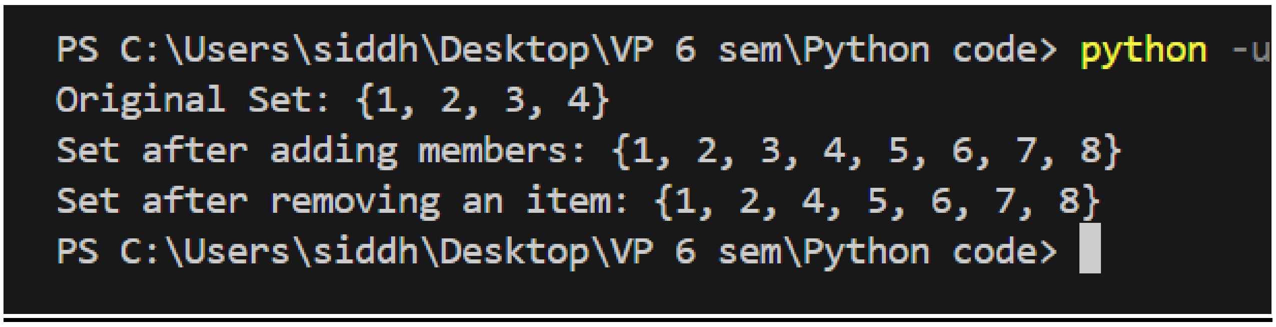
1. Write a Python program to create a set, add member(s) in a set and remove one item from set..

Ans:

a] Code:

```
my_set = {1, 2, 3, 4}

print("Original Set:", my_set)

my_set.add(5)

my_set.update([6, 7, 8])

print("Set after adding members:", my_set)

my_set.remove(3)

print("Set after removing an item:", my_set)
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -u
Original Set: {1, 2, 3, 4}
Set after adding members: {1, 2, 3, 4, 5, 6, 7, 8}
Set after removing an item: {1, 2, 4, 5, 6, 7, 8}
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

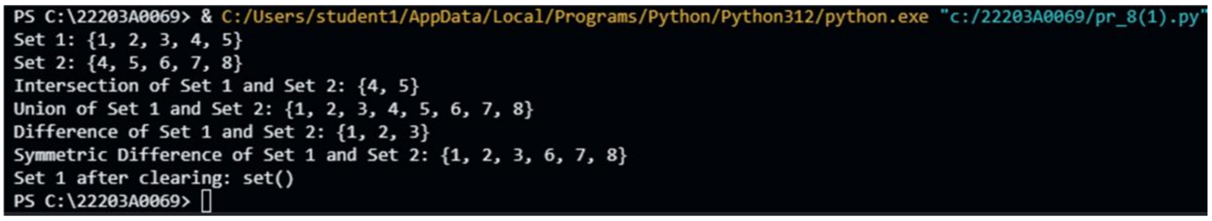
2. Write a Python program to perform following operations on set: intersection of sets, union of sets, set difference, symmetric difference, clear a set.

Ans:

a] Code:

```
set1 = {1, 2, 3, 4, 5}

set2 = {4, 5, 6, 7, 8}

print("Set 1:", set1)

print("Set 2:", set2) intersection = set1 & set2

print("Intersection of Set 1 and Set 2:", intersection) union = set1 | set2

print("Union of Set 1 and Set 2:", union) difference = set1 - set2

print("Difference of Set 1 and Set 2:", difference) symmetric_difference = set1 ^ set2

print("Symmetric Difference of Set 1 and Set 2:", symmetric_difference) set1.clear()

print("Set 1 after clearing:", set1)
```

b] Output:



```
PS C:\22203A0069> & C:/Users/student1/AppData/Local/Programs/Python/Python312/python.exe "c:/22203A0069/pr_8(1).py"
Set 1: {1, 2, 3, 4, 5}
Set 2: {4, 5, 6, 7, 8}
Intersection of Set 1 and Set 2: {4, 5}
Union of Set 1 and Set 2: {1, 2, 3, 4, 5, 6, 7, 8}
Difference of Set 1 and Set 2: {1, 2, 3}
Symmetric Difference of Set 1 and Set 2: {1, 2, 3, 6, 7, 8}
Set 1 after clearing: set()
PS C:\22203A0069> 
```

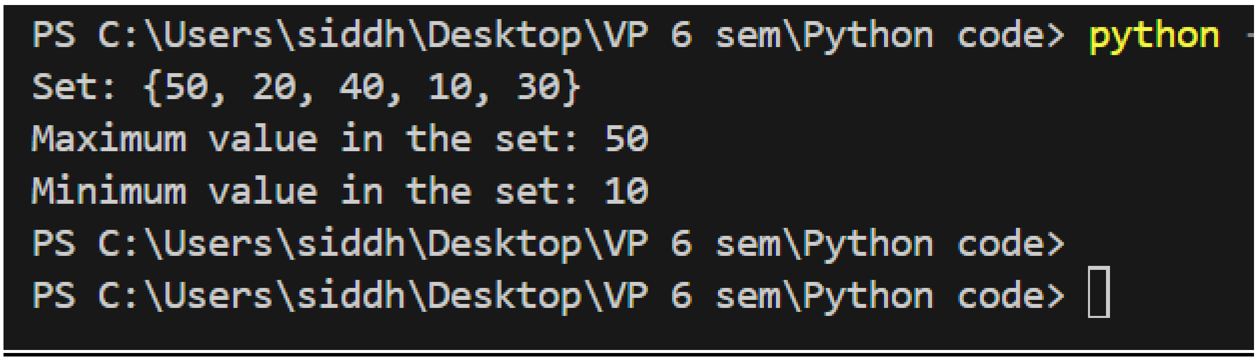
3. Write a Python program to find maximum and the minimum value in a set.

Ans:

<u>a] Code:</u>

```
my_set = {10, 20, 30, 40, 50}

max_value = max(my_set)

min_value = min(my_set)

print("Set:", my_set)

print("Maximum value in the set:", max_value)

print("Minimum value in the set:", min_value)
```

<u>b] Output:</u>



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python 
Set: {50, 20, 40, 10, 30}
Maximum value in the set: 50
Minimum value in the set: 10
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> 
```

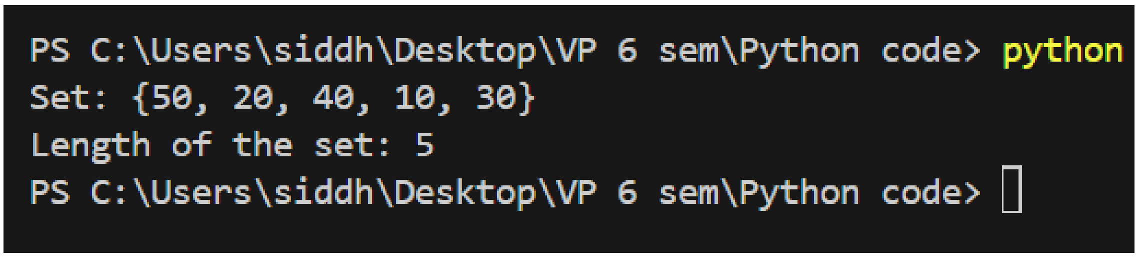
4. Write a Python program to find the length of a set

Ans:

a] Code:

```
my_set = {10, 20, 30, 40, 50}

length_of_set = len(my_set)

print("Set:", my_set)

print("Length of the set:", length_of_set)
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python
Set: {50, 20, 40, 10, 30}
Length of the set: 5
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> 
```

Vidyalankar Polytechnic	DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	09
Title of Experiment	Write Python program to perform following operations on Dictionaries: Create Dictionary, Access Dictionary elements, Update Dictionary, Delete Dictionary, Looping through Dictionary.

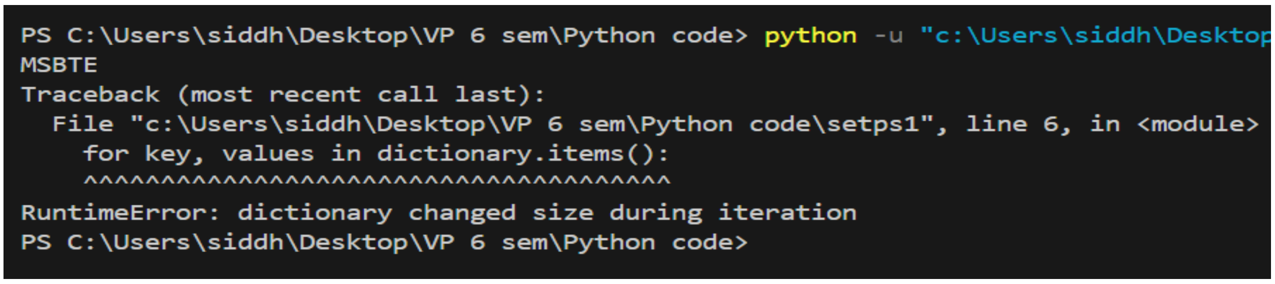
PRACTICAL RELATED QUESTIONS:

1. What is the output of the following program?

```
dictionary = {'MSBTE' : 'Maharashtra State Board of Technical Education',
'CO' : 'Computer Engineering',
'SEM' : 'Sixth'
}
del dictionary['CO']
for key, values in dictionary.items():
    print(key)
    dictionary.clear()
for key, values in dictionary.items():
    print(key)
    del dictionary
for key, values in dictionary.items():
    print(key)
```

Ans:

Output:



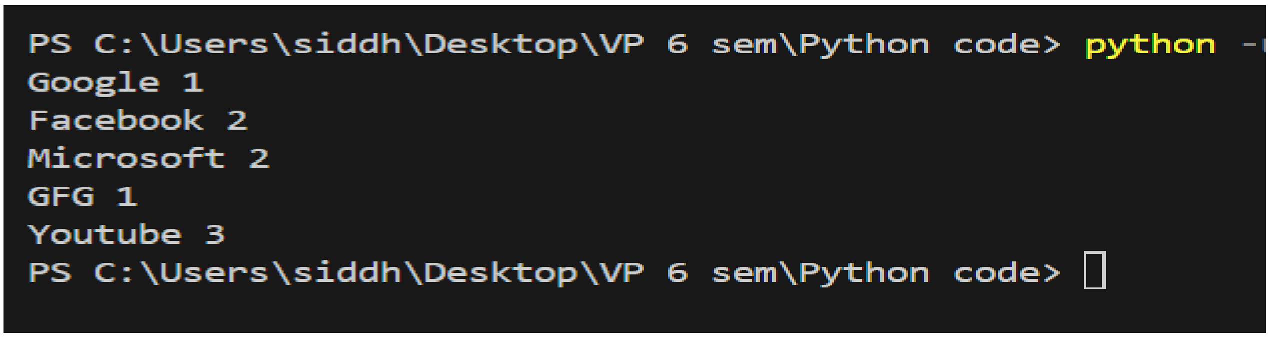
```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -u "c:\Users\siddh\Desktop
MSBTE
Traceback (most recent call last):
  File "c:\Users\siddh\Desktop\VP 6 sem\Python code\setps1", line 6, in <module>
    for key, values in dictionary.items():
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
RuntimeError: dictionary changed size during iteration
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

2. What is the output of the following program?

```
 dictionary1 = {'Google' : 1,
 'Facebook' : 2,
 'Microsoft' : 3
 }
dictionary2 = {'GFG' : 1,
 'Microsoft' : 2,
 'Youtube' : 3
 }
dictionary1.update(dictionary2);
for key, values in dictionary1.items():
    print(key, values)
```

Ans:

Output:



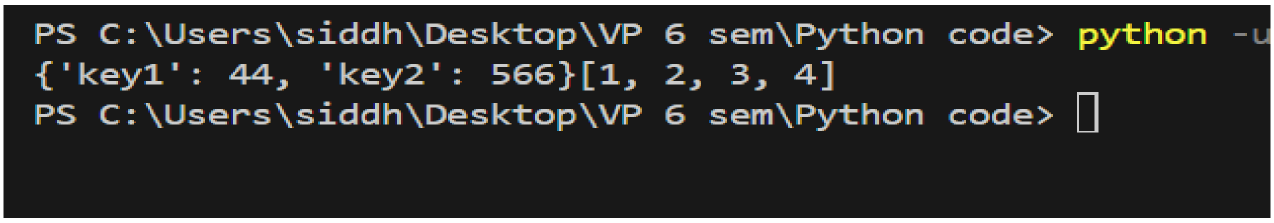
```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -
Google 1
Facebook 2
Microsoft 2
GFG 1
Youtube 3
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> 
```

3. What is the output of the following program?

```
temp = dict()
temp['key1'] = {'key1' : 44, 'key2' : 566}
temp['key2'] = [1, 2, 3, 4]
for (key, values) in temp.items():
    print(values, end = "")
```

Ans:

Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -u
{'key1': 44, 'key2': 566}[1, 2, 3, 4]
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> 
```

EXERCISE:

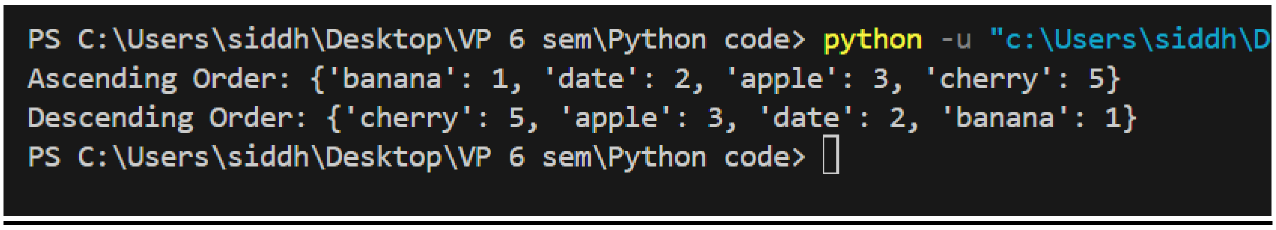
1. Write a Python script to sort (ascending and descending) a dictionary by value

Ans:

a] Code:

```
data = {'apple': 3, 'banana': 1, 'cherry': 5, 'date': 2}

ascending = sorted(data.items(), key=lambda x: x[1])

descending = sorted(data.items(), key=lambda x: x[1], reverse=True)

print("Ascending Order:", dict(ascending))

print("Descending Order:", dict(descending))
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -u "c:\Users\siddh\D
Ascending Order: {'banana': 1, 'date': 2, 'apple': 3, 'cherry': 5}
Descending Order: {'cherry': 5, 'apple': 3, 'date': 2, 'banana': 1}
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

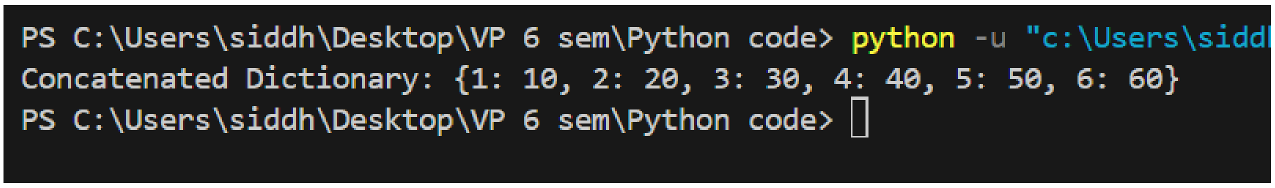
2. Write a Python script to concatenate following dictionaries to create a new one.
a. Sample Dictionary:
b. dic1 = {1:10, 2:20}
c. dic2 = {3:30, 4:40}
d. dic3 = {5:50,6:60}

Ans:

a] Code:

```
dic1 = {1: 10, 2: 20}

dic2 = {3: 30, 4: 40}

dic3 = {5: 50, 6: 60}

new_dict = {}

new_dict.update(dic1)

new_dict.update(dic2)

new_dict.update(dic3)

print("Concatenated Dictionary:", new_dict)
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -u "c:\Users\siddh
Concatenated Dictionary: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

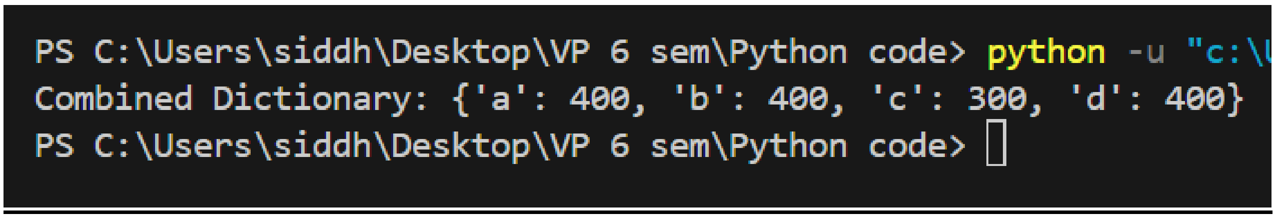
3. Write a Python program to combine two dictionary adding values for common keys.
a. d1 = {'a': 100, 'b': 200, 'c':300}
b. d2 = {'a': 300, 'b': 200, 'd':400}

Ans:

<u>a] Code:</u>

```
d1 = {'a': 100, 'b': 200, 'c': 300}

d2 = {'a': 300, 'b': 200, 'd': 400}

combined_dict = d1.copy()

for key, value in d2.items():

    if key in combined_dict:

        combined_dict[key] += value

    else:

        combined_dict[key] = value

print("Combined Dictionary:", combined_dict)
```

<u>b] Output:</u>



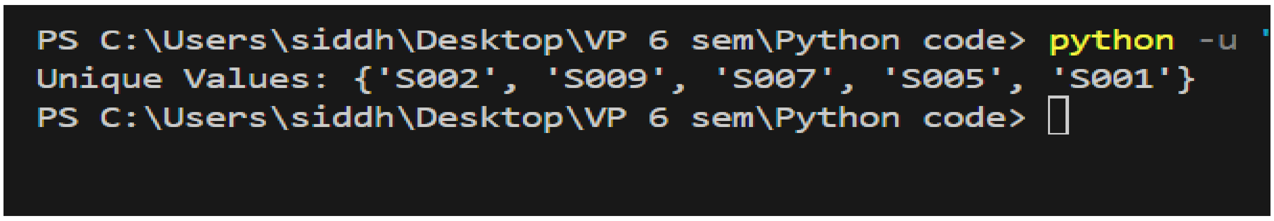
```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -u "c:\
Combined Dictionary: {'a': 400, 'b': 400, 'c': 300, 'd': 400}
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> 
```

4. Write a Python program to print all unique values in a dictionary.
a. Sample Data: [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII":"S005"}, {"V":"S009"}, {"VIII":"S007"}]
Ans:

a] Code:

```
data = [{"V": "S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"},{"VII": "S005"}, {"V": "S009"}, {"VIII": "S007"}]

unique_values = set()

for d in data:

    for value in d.values():

        unique_values.add(value)

print("Unique Values:", unique_values)
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -u '
Unique Values: {'S002', 'S009', 'S007', 'S005', 'S001'}
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

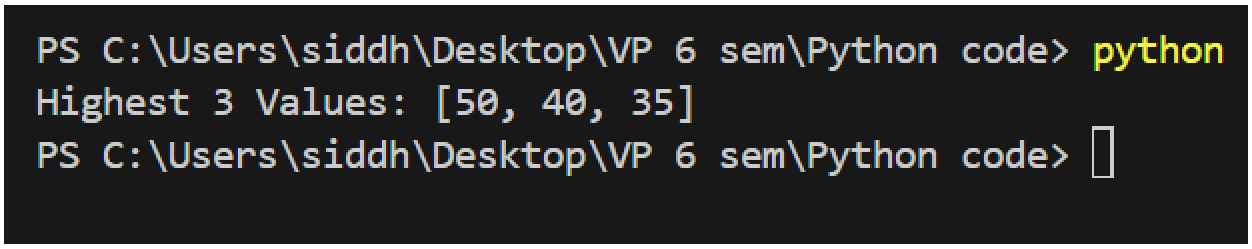
5. Write a Python program to find the highest 3 values in a dictionary.

Ans:

a] Code:

```
data = {'a': 10, 'b': 25, 'c': 35, 'd': 50, 'e': 40}

highest_3_values = sorted(data.values(), reverse=True)[:3]

print("Highest 3 Values:", highest_3_values)
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python
Highest 3 Values: [50, 40, 35]
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

Vidyalankar Polytechnic

DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	10
Title of Experiment	Write Python program to demonstrate math built-in functions (Any 2 programs) Write Python program to demonstrate string built-in functions (Any 2 programs)

X. Practical related Questions

1. Describe about string formatting operator with example.

ANS:

Specifier	**Description**
%d	Integer
%f	Floating point number
%s	String
%x	Hexadecimal
%o	Octal

2. Give the syntax and example of title() and capitalize() methods

Ans:

1. title() Method

The title() method converts the first character of each word in a string to uppercase and the rest to lowercase.

Syntax:

```
string.title()
```

Example:

```
text = "hello world! welcome to python."
print(text.title())
```

Output:

```
Hello World! Welcome To Python.
```

2. capitalize() Method

The capitalize() method converts the **first character** of the string to uppercase and the rest to lowercase.

Syntax:

```
string.capitalize()
```

Example:

```
text = "hello world! welcome to python."
print(text.capitalize())
```

Output:

```
Hello world! welcome to python.
```

3. Give the syntax and significance of string functions: title() and strip()

Ans:

1. title() Method

Syntax:

```
string.title()
```

Significance:

- Converts the first letter of each word in the string to uppercase and the rest to lowercase.
- Useful for formatting text, such as names and titles.

Example:

```
text = "hello world! welcome to python."

print(text.title())
```

Output:

```
Hello World! Welcome To Python.
```

2. strip() Method

Syntax:

```
string.strip()
```

Significance:

- Removes **leading and trailing** whitespace (spaces, newlines, tabs) from a string.
- Useful for cleaning user input or removing unwanted spaces in data.

Example:

```
text = "   Hello World!   "

print(text.strip())
```

Output:

```
Hello World!
```

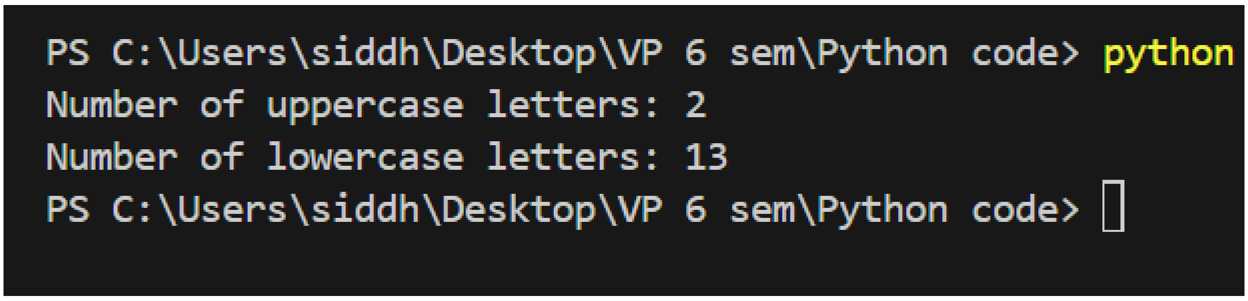
XI. Exercise

1. Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.
Ans:

a] Code:

```
text = "Welcome to Python"

upper = 0

lower = 0

for i in range(len(text)):

    if text[i].isupper():

        upper += 1

    elif text[i].islower():

        lower += 1

print("Number of uppercase letters:", upper)

print("Number of lowercase letters:", lower)
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python
Number of uppercase letters: 2
Number of lowercase letters: 13
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

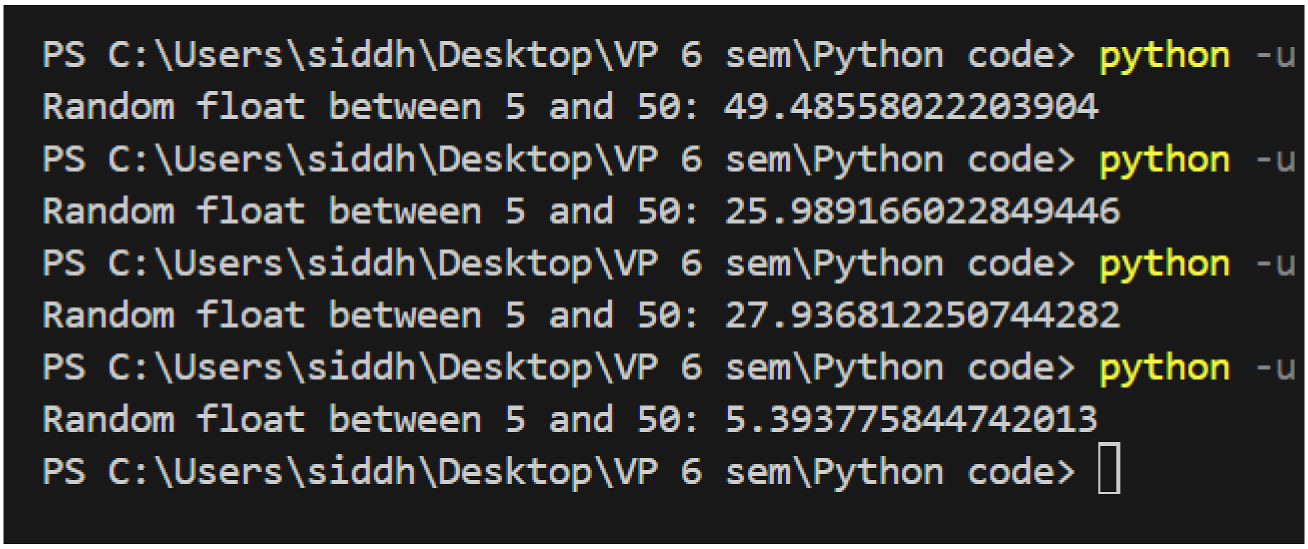
2. Write a Python program to generate a random float where the value is between 5 and 50 using Python math module.

Ans:

a] Code:

```
import random

random_float = 5 + (random.random() * (50 - 5))

print("Random float between 5 and 50:", random_float)
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -u
Random float between 5 and 50: 49.48558022203904
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -u
Random float between 5 and 50: 25.989166022849446
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -u
Random float between 5 and 50: 27.936812250744282
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -u
Random float between 5 and 50: 5.393775844742013
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> 
```

Vidyalankar Polytechnic	DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	11
Title of Experiment	Develop user defined Python function for given problem: Function with minimum 2 arguments Function returning values

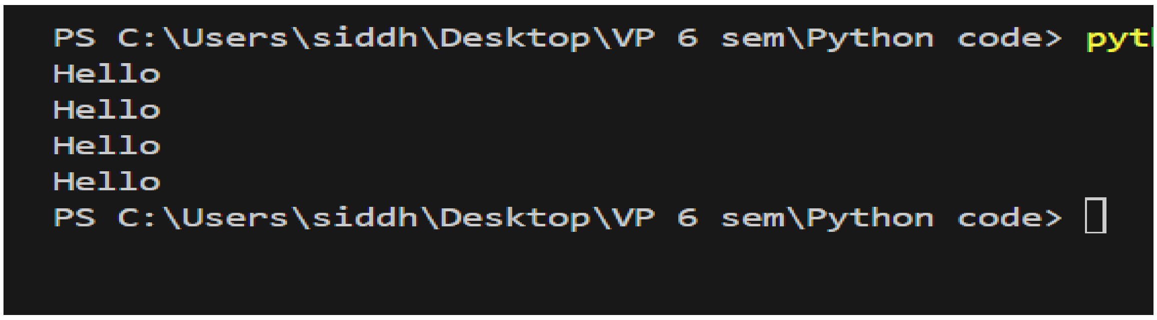
X. Practical related Questions

1. What is the output of the following program?

```
def myfunc(text, num):
        while num > 0:
                print(text)
                num = num - 1
myfunc('Hello', 4)
```

ANS:

Output:



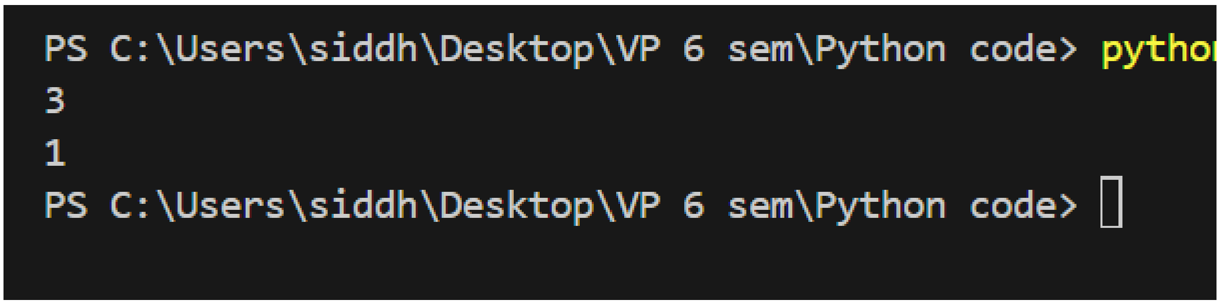
```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> pyt
Hello
Hello
Hello
Hello
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

2. What is the output of the following program?

```
num = 1
def func():
        num = 3
        print(num)
func()
print(num)
```

Ans:

Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> pytho
3
1
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

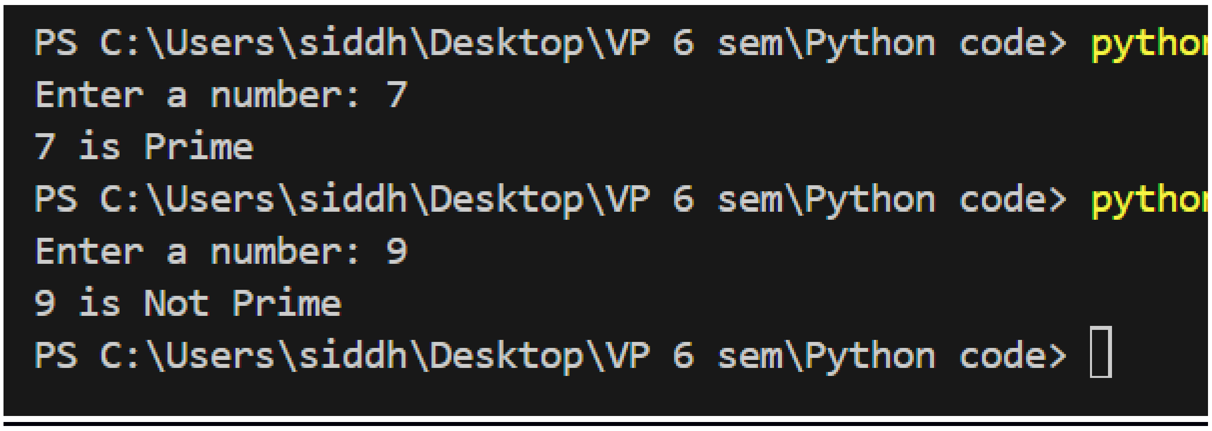
XI. Exercise

1. Write a Python function that takes a number as a parameter and check the number is prime or not.
Ans:

a] Code:

```
num = int(input("Enter a number: "))
if num <= 1:
    print("Neither Prime Nor Composite")
else:
    flag = 0
    for i in range(2, num):
        if num % i == 0:
            flag = 1
            break
    if flag == 0:
        print(num, "is Prime")
    else:
        print(num, "is Not Prime")
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> pytho
Enter a number: 7
7 is Prime
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> pytho
Enter a number: 9
9 is Not Prime
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> 
```

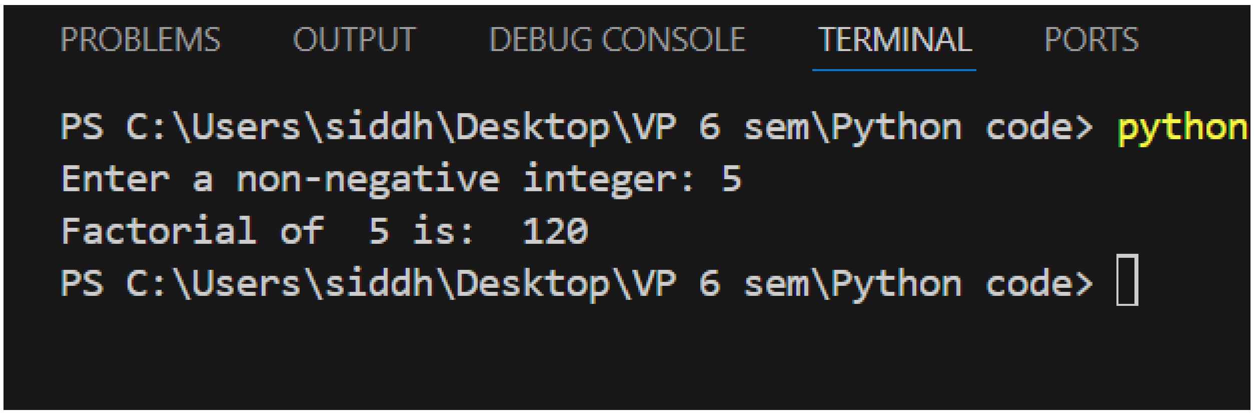
2. Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument

Ans:

a] Code:

```
num = int(input("Enter a non-negative integer: "))

def factorial(n):

    result = 1

    for i in range(2, n + 1):

        result *= i

    return result

print("Factorial of ",num,"is: ",factorial(num))
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python
Enter a non-negative integer: 5
Factorial of  5 is:  120
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> 
```

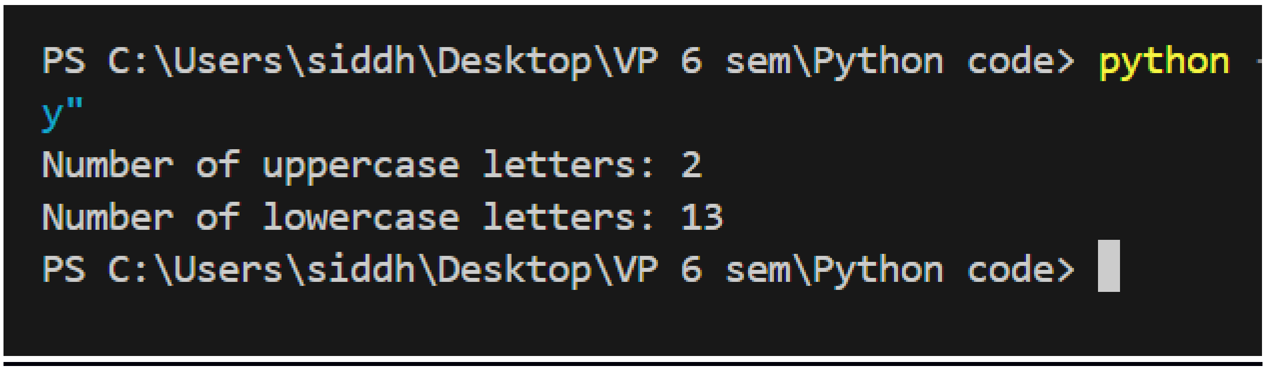
3. Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.

Ans:

a] Code:

```
text = "Welcome to Python"

upper = 0

lower = 0

for i in range(len(text)):

    if text[i].isupper():

        upper += 1

    elif text[i].islower():

        lower += 1

print("Number of uppercase letters:", upper)

print("Number of lowercase letters:", lower)
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python
y"
Number of uppercase letters: 2
Number of lowercase letters: 13
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	12
Title of Experiment	Write Python program to demonstrate use of: a) Built in module (e.g. keyword, math, number, operator) b) user defined module.

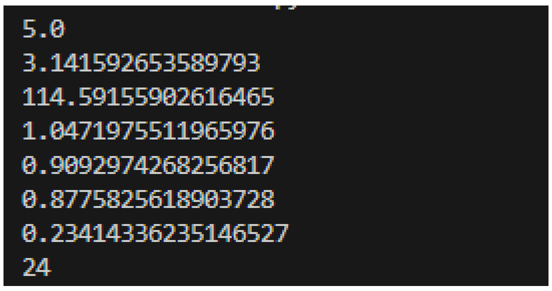
X. PRACTICAL RELATED QUESTIONS

1. What is the output of the following program?

```
import math
print (math.sqrt(25))
print (math.pi)
print (math.degrees(2))
print (math.radians(60))
print (math.sin(2))
print (math.cos(0.5))
print (math.tan(0.23))
print (math.factorial(4))
```

ANS:

Output-



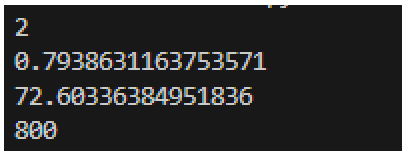
```
5.0
3.141592653589793
114.59155902616465
1.0471975511965976
0.9092974268256817
0.8775825618903728
0.23414336235146527
24
```

2. What is the output of the following program?

```
import random


print (random.randint(0, 5))

print (random.random())

print (random.random() * 100)

List = [1, 4, True, 800, "Python", 27, "hello"]

print (random.choice(List) )
```

Ans:

Output-

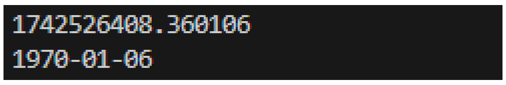


```
2
0.7938631163753571
72.60336384951836
800
```

3. What is the output of the following program?

```
import datetime

from datetime import date

import time


print (time.time())

print (date.fromtimestamp(454554))
```

Ans:



```
1742526408.360106
1970-01-06
```

XI. Exercise

1. **Write a Python program to create a user defined module that will ask your college name and will display the name of the college ?**

Ans:

Code:

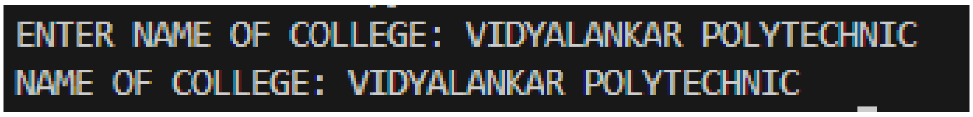
college.py-

```
def setName(name):
    print("NAME OF COLLEGE: "+name)
```

mainfile.py-

```
from college import *


setName(input("ENTER NAME OF COLLEGE: "))
```

Output:



```
ENTER NAME OF COLLEGE: VIDYALANKAR POLYTECHNIC
NAME OF COLLEGE: VIDYALANKAR POLYTECHNIC
```

2. Write a Python program that will calculate area and circumference of circle using inbuilt Math Module ?

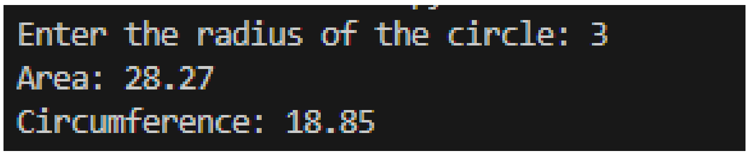
Ans:

Code:

```
import math


radius = float(input("Enter the radius of the circle: "))


area = math.pi * radius ** 2
```

```
circumference = 2 * math.pi * radius


print(f"Area: {area:.2f}")

print(f"Circumference: {circumference:.2f}")
```

Output:



```
Enter the radius of the circle: 3
Area: 28.27
Circumference: 18.85
```

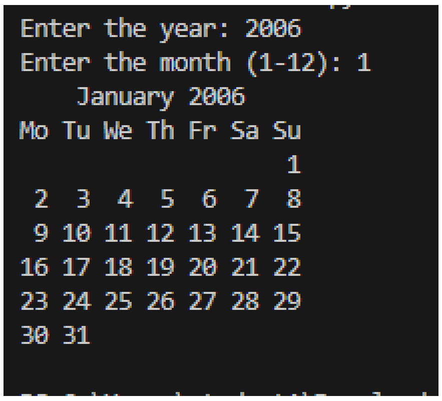
3. Write a Python program that will display Calendar of given month using Calendar Module ?

Ans:

Code:

```
import calendar


year = int(input("Enter the year: "))

month = int(input("Enter the month (1-12): "))


print(calendar.month(year, month))
```

Output:



```
Enter the year: 2006
Enter the month (1-12): 1
    January 2006
Mo Tu We Th Fr Sa Su
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

Vidyalankar Polytechnic	DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code: 22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	13
Title of Experiment	Write Python program to demonstrate use of: • built-in packages (e.g. NumPy, Pandas) • user defined packages

X. PRACTICAL RELATED QUESTIONS

1. Describe Numpy Array ?

ANS:

- A NumPy array (ndarray) is a powerful, multi-dimensional array object provided by the NumPy library in Python.
- It is faster and more efficient than Python lists because it supports vectorized operations and is implemented in C.
- Supports homogeneous data types (all elements must be of the same type).
- Provides built-in mathematical functions, making it ideal for numerical and scientific computing.

2. Why should we use Numpy rather than Matlab, Octave or Yorick?

Ans:

- Open Source & Free: Unlike MATLAB, NumPy is open-source and free to use.
- Python Integration: Seamlessly integrates with Python, which is widely used in data science, ML, and AI.
- Performance: Faster execution due to efficient memory management and vectorized operations.
- Extensive Libraries: Can be used alongside Pandas, SciPy, TensorFlow, and Scikit-learn for advanced applications.
- Better Interoperability: Works well with other programming languages like C, C++, and Fortran.

XI. Exercise

1. **Write a Python program to create two matrices and perform addition, subtraction, multiplication and division operation on matrix ?**

Ans:

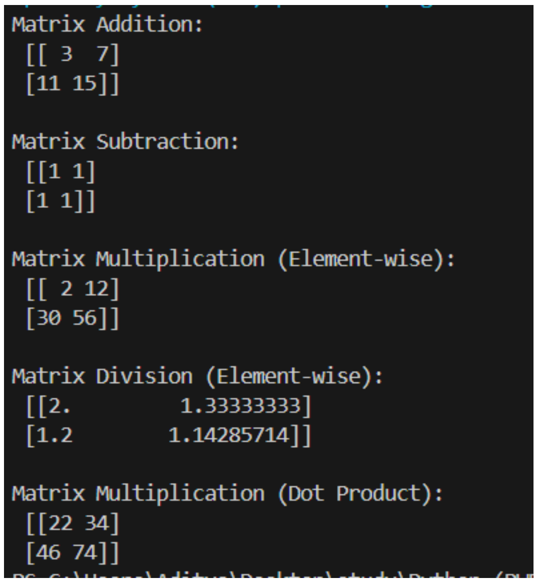
Code:

```
import numpy as np


# Creating two matrices

A = np.array([[2, 4], [6, 8]])

B = np.array([[1, 3], [5, 7]])


# Matrix Addition

addition = A + B

print("Matrix Addition:\n", addition)


# Matrix Subtraction

subtraction = A - B

print("\nMatrix Subtraction:\n", subtraction)


# Matrix Multiplication (Element-wise)

multiplication = A * B

print("\nMatrix Multiplication (Element-wise):\n", multiplication)


# Matrix Division (Element-wise)

division = A / B

print("\nMatrix Division (Element-wise):\n", division)


# Matrix Multiplication (Dot Product)
```

dot_product = np.dot(A, B)

print("\nMatrix Multiplication (Dot Product):\n", dot_product)

Output:



```
Matrix Addition:
 [[ 3  7]
 [11 15]]

Matrix Subtraction:
 [[1 1]
 [1 1]]

Matrix Multiplication (Element-wise):
 [[ 2 12]
 [30 56]]

Matrix Division (Element-wise):
 [[2.         1.33333333]
 [1.2        1.14285714]]

Matrix Multiplication (Dot Product):
 [[22 34]
 [46 74]]
```

2. Write a Python program to concatenate two strings ?

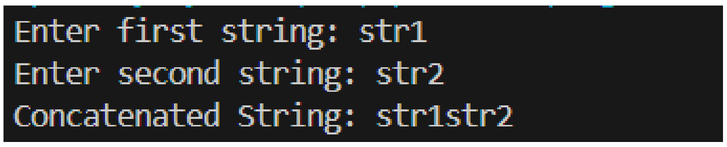
Ans:

Code:

```
from string import Template


# Taking input from the user

str1 = input("Enter first string: ")

str2 = input("Enter second string: ")


# Concatenating the strings using Template

template = Template("$a$b")

concatenated_str = template.substitute(a=str1, b=str2)


# Displaying the result
```

print("Concatenated String:", concatenated_str)

Output:



```
Enter first string: str1
Enter second string: str2
Concatenated String: str1str2
```

3. Write a NumPy program to generate six random integers between 10 and 30 ?

Ans:

Code:

import numpy as np

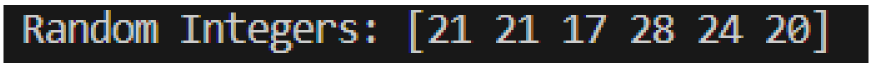
# Generate six random integers between 10 and 30

random_numbers = np.random.randint(10, 31, size=6)

# Display the random numbers

print("Random Integers:", random_numbers)

Output:



```
Random Integers: [21 21 17 28 24 20]
```

Vidyalankar Polytechnic	DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code: 22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	14
Title of Experiment	Write a program in Python to demonstrate following operations: (a) Method overloading (b) Method overriding

X. PRACTICAL RELATED QUESTIONS

1. State the difference between method overriding and overloading ?

ANS:

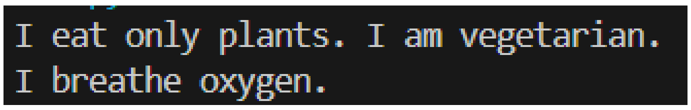
Feature	**Method Overloading**	**Method Overriding**
Definition	**Defining multiple methods with the same name but different parameters within the same class.**	**Redefining a method from the parent class in the child class with the same name and parameters.**
Number of Classes	**Happens within a single class.**	**Occurs between a parent class and a child class (inheritance).**
Parameters	**Methods differ in the number or type of parameters.**	**Method signatures (name & parameters) remain the same in both classes.**
Usage	**Provides multiple ways to call a method based on different arguments.**	**Modifies or extends the behavior of a method inherited from a parent class.**
Python Support	**Python does not support true method overloading, but it can be achieved using default parameters or *args/**kwargs.**	**Fully supported using inheritance and method overriding in the child class.**

2. What is the output of the following program?

```
# parent class
class Animal:
    # properties
    multicellular = True
    # Eukaryotic means Cells with Nucleus
    eukaryotic = True
    # function breath
    def breathe(self):
        print("I breathe oxygen.")
    # function feed
    def feed(self):
        print("I eat food.")


# child class
class Herbivorous(Animal):
    # function feed
    def feed(self):
        print("I eat only plants. I am vegetarian.")


herbi = Herbivorous()
herbi.feed()
# calling some other function
herbi.breathe()
```

Ans:

Output-



```
I eat only plants. I am vegetarian.
I breathe oxygen.
```

XI. Exercise

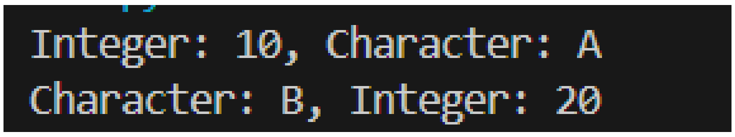
1. **Write a Python program to create a class to print an integer and a character with two methods having the same name but different sequence of the integer and the character parameters. For example, if the parameters of the first method are of the form (int n, char c), then that of the second method will be of the form (char c, int n) ?**

Ans:

Code:

```
class PrintValues:

    def display(self, a, b):

        if isinstance(a, int) and isinstance(b, str):

            print(f"Integer: {a}, Character: {b}")

        elif isinstance(a, str) and isinstance(b, int):

            print(f"Character: {a}, Integer: {b}")

        else:

            print("Invalid input types!")


obj = PrintValues()


obj.display(10, 'A')

obj.display('B', 20)
```

Output:



```
Integer: 10, Character: A
Character: B, Integer: 20
```

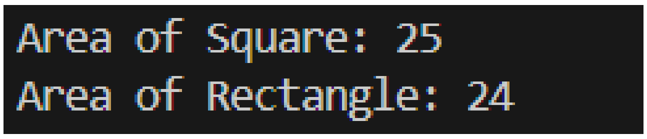
2. Write a Python program to create a class to print the area of a square and a rectangle. The class has two methods with the same name but different number of parameters. The method for printing area of rectangle has two parameters which are length and breadth respectively while the other method for printing area of square has one parameter which is side of square ?

Ans:

Code:

```
class AreaCalculator:
    def area(self, a, b=None):
        if b is None:
            print(f"Area of Square: {a * a}")
        else:
            print(f"Area of Rectangle: {a * b}")


obj = AreaCalculator()


obj.area(5)
obj.area(4, 6)
```

Output:



```
Area of Square: 25
Area of Rectangle: 24
```

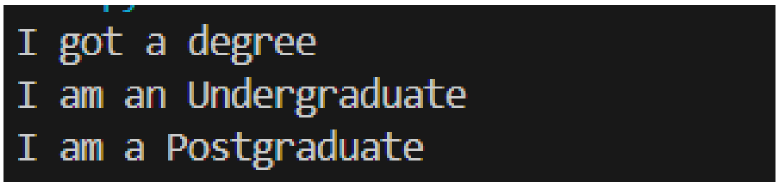
3. Write a Python program to create a class 'Degree' having a method 'getDegree' that prints "I got a degree". It has two subclasses namely 'Undergraduate' and 'Postgraduate' each having a method with the same name that prints "I am an Undergraduate" and "I am a Postgraduate" respectively. Call the method by creating an object of each of the three classes ?

Ans:

Code:

```
class Degree:
    def getDegree(self):
        print("I got a degree")


class Undergraduate(Degree):
    def getDegree(self):
        print("I am an Undergraduate")
class Postgraduate(Degree):
    def getDegree(self):
        print("I am a Postgraduate")
# Creating objects of each class
degree = Degree()
undergrad = Undergraduate()
postgrad = Postgraduate()
# Calling the getDegree method for each object
degree.getDegree()
undergrad.getDegree()
postgrad.getDegree()
```

Output:



```
I got a degree
I am an Undergraduate
I am a Postgraduate
```

Vidyalankar Polytechnic	DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	15
Title of Experiment	Write a program in Python to demonstrate following operations: Simple inheritance, Multiple inheritance

X. Practical related Questions

1. State the use of inheritance

Ans:

In Python, inheritance allows you to create new classes (child classes) that inherit attributes and methods from existing classes (parent classes), promoting code reusability and hierarchical structures

2. List different types of inheritance

Ans:

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance

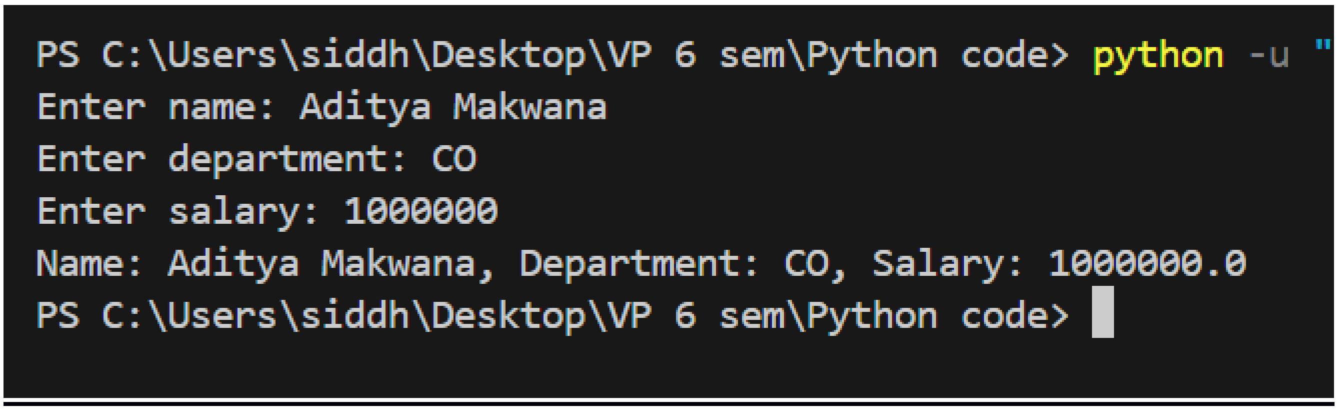
XI. Exercise

1. Create a class Employee with data members: name, department and salary. Create suitable methods for reading and printing employee information
Ans:

a] Code:

```
class Employee:

    def display_info(self):

        print(f"Name: {self.name}, Department: {self.department}, Salary: {self.salary}")

    def input_info(self):

        self.name = input("Enter name: ")

        self.department = input("Enter department: ")

        self.salary = float(input("Enter salary: "))

employee = Employee()

employee.input_info()

employee.display_info()
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -u "
Enter name: Aditya Makwana
Enter department: CO
Enter salary: 1000000
Name: Aditya Makwana, Department: CO, Salary: 1000000.0
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> 
```

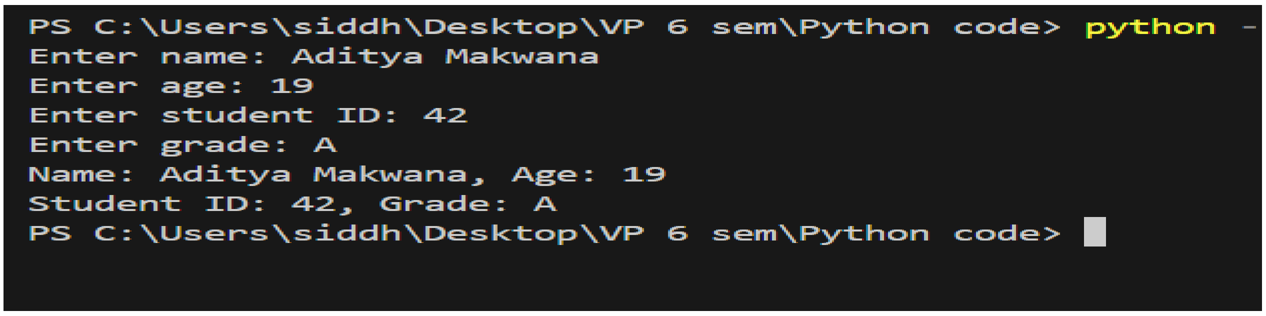
2. Python program to read and print students information using two classes using simple inheritance.

Ans:

a] Code:

```
class Person:
    def input_info(self):
        self.name = input("Enter name: ")
        self.age = int(input("Enter age: "))

    def display_info(self):
        print(f"Name: {self.name}, Age: {self.age}")

class Student(Person):
    def input_info(self):
        super().input_info()
        self.student_id = input("Enter student ID: ")
        self.grade = input("Enter grade: ")
    def display_info(self):
        super().display_info()
        print(f"Student ID: {self.student_id}, Grade: {self.grade}")
student = Student()
student.input_info()
student.display_info()
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python -
Enter name: Aditya Makwana
Enter age: 19
Enter student ID: 42
Enter grade: A
Name: Aditya Makwana, Age: 19
Student ID: 42, Grade: A
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

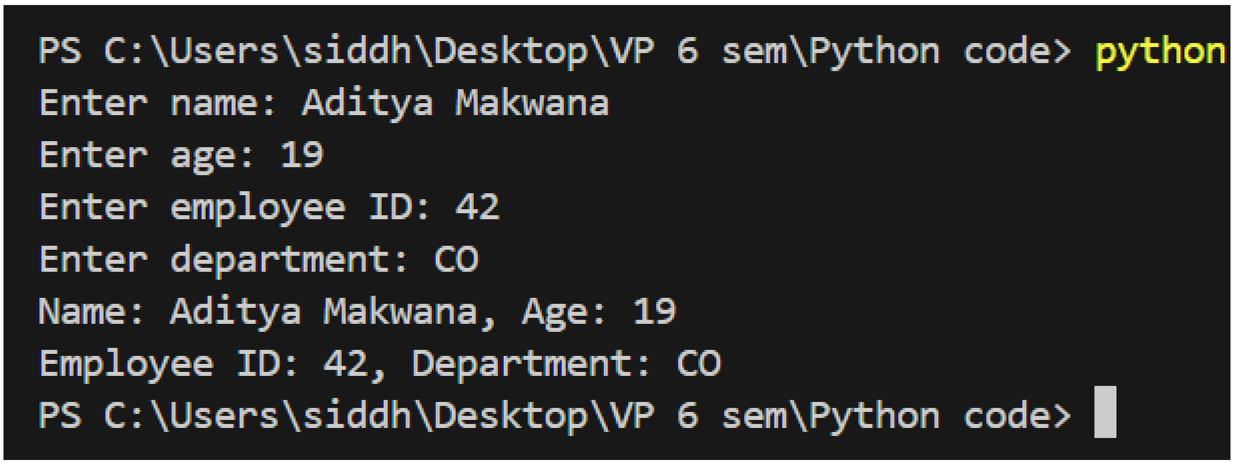
3. Write a Python program to implement multiple inheritance

Ans:

a] Code:

```
class Person:
    def input_person_info(self):
        self.name = input("Enter name: ")
        self.age = int(input("Enter age: "))

    def display_person_info(self):
        print(f"Name: {self.name}, Age: {self.age}")
class Employee:
    def input_employee_info(self):
        self.employee_id = input("Enter employee ID: ")
        self.department = input("Enter department: ")

    def display_employee_info(self):
        print(f"Employee ID: {self.employee_id}, Department: {self.department}")
class Manager(Person, Employee):
    def input_info(self):
        self.input_person_info()
        self.input_employee_info()
    def display_info(self):
        self.display_person_info()
        self.display_employee_info()
manager = Manager()
manager.input_info()
manager.display_info()
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python
Enter name: Aditya Makwana
Enter age: 19
Enter employee ID: 42
Enter department: CO
Name: Aditya Makwana, Age: 19
Employee ID: 42, Department: CO
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

DEPARTMENT OF COMPUTER ENGINEERING

Subject: Programming with Python	Subject Code:22616
Semester:6th Semester	Course: Computer Engineering
Laboratory No: L001B	Name of Subject Teacher: Sangeeta Shirsat
Name of Student: Aditya Makwana	Roll Id: 22203A0042

Experiment No:	16
Title of Experiment	Write a program in Python to handle user defined exception for given problem

X. Practical related Questions

1. State Exception.

Ans:

Python Exception Handling handles errors that occur during the execution of a program. Exception handling allows to respond to the error, instead of crashing the running program. It enables you to catch and manage errors, making your code more robust and user-friendly.

2. How to handle exception in Python

Ans:

In Python, you handle exceptions (runtime errors) using try and except blocks, allowing you to catch and manage errors gracefully instead of crashing the program

Example:

```
try:
    # Code that might raise an exception
    result = 10 / 0  # This will raise a ZeroDivisionError
except ZeroDivisionError:
    print("Error: Division by zero!")
else:
    print("No exceptions occurred.")
finally:
    print("This code always executes.")
```

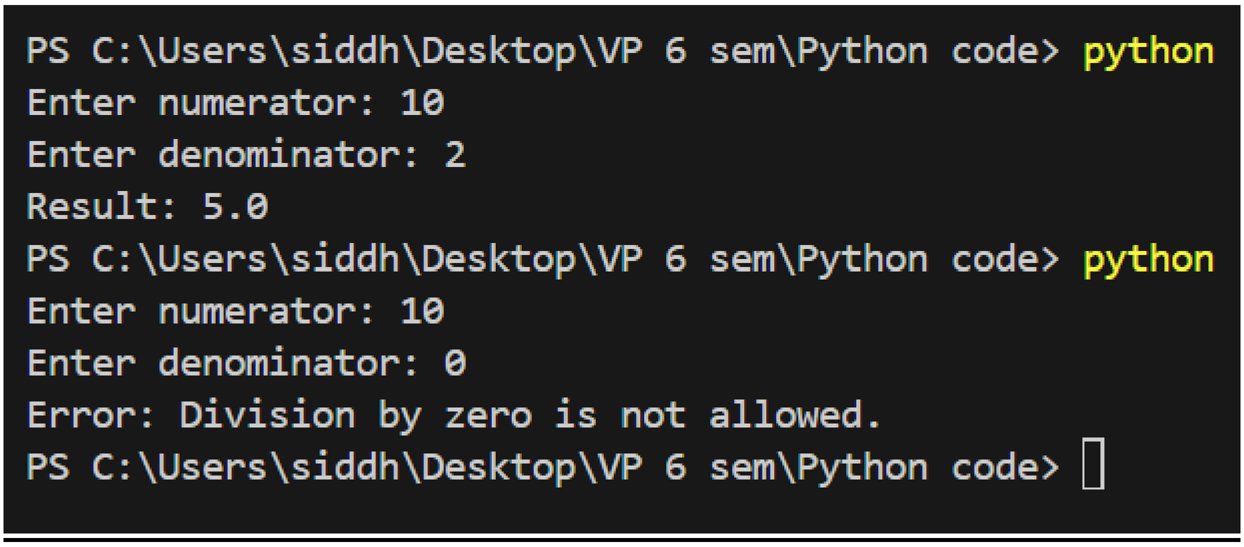
XI. Exercise

1. Write a Python program to Check for ZeroDivisionError Exception.
Ans:

a] Code:

```
try:
    numerator = float(input("Enter numerator: "))
    denominator = float(input("Enter denominator: "))
    result = numerator / denominator
    print(f"Result: {result}")
except ZeroDivisionError:
    print("Error: Division by zero is not allowed.")
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python
Enter numerator: 10
Enter denominator: 2
Result: 5.0
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python
Enter numerator: 10
Enter denominator: 0
Error: Division by zero is not allowed.
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```

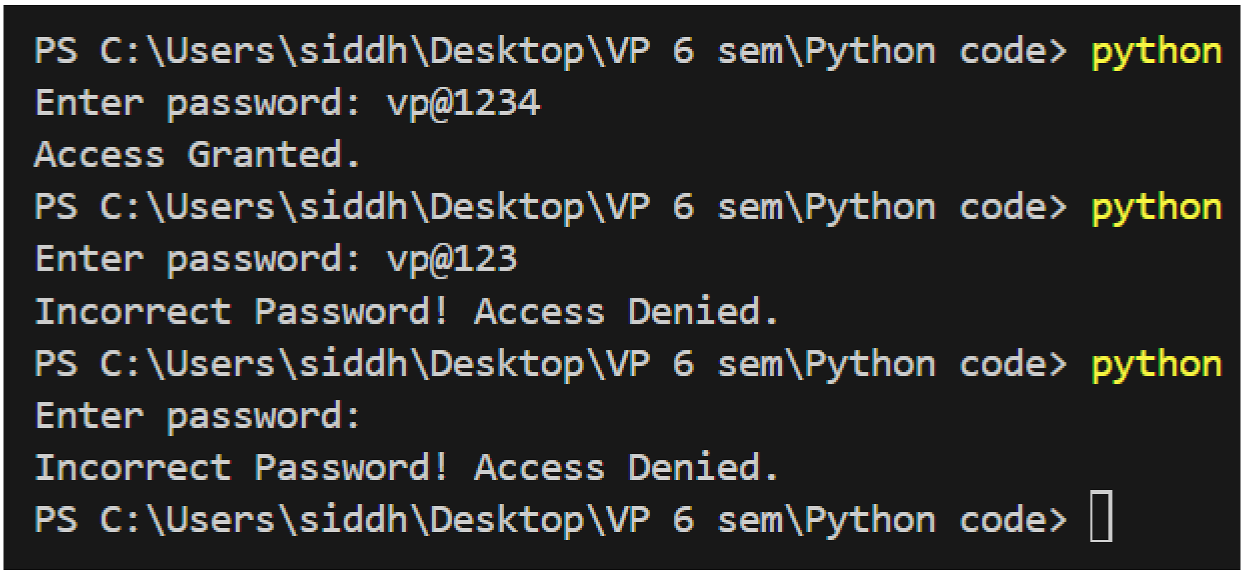
2. Write a Python program to create user defined exception that will check whether the password is correct or not?

Ans:

a] Code:

```
class InvalidPasswordError(Exception):
    pass
def check_password(password):
    correct_password = "vp@1234"
    if password != correct_password:
        raise InvalidPasswordError("Incorrect Password! Access Denied.")
    else:
        print("Access Granted.")


try:
    user_password = input("Enter password: ")
    check_password(user_password)
except InvalidPasswordError as e:
    print(e)
```

b] Output:



```
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python
Enter password: vp@1234
Access Granted.
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python
Enter password: vp@123
Incorrect Password! Access Denied.
PS C:\Users\siddh\Desktop\VP 6 sem\Python code> python
Enter password:
Incorrect Password! Access Denied.
PS C:\Users\siddh\Desktop\VP 6 sem\Python code>
```