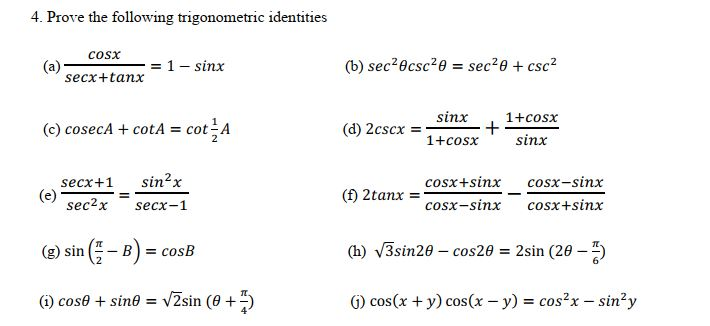
**Lab Session #2**

**Exploring Math Module in Python**

**Aim:** To Learn Math module in python and Develop Programs To Solve The Given Mathematical Problems.

**Problem Definition:** Develop Python Programs to do the following:

1. Evaluate where x is taken from the user.
2. Evaluate where x is taken from the user.
3. Evaluated where x is taken from the user.
4. Evaluate Sin, cos, tan of angles read from the user.
5. Find the absolute value of x
6. Convert from degrees to radians and vice versa using functions.
7. Prove the following identities:



**Theory**: The Python math module is an important feature designed to deal with mathematical operations. It comes packaged with the standard Python release and has been there from the beginning. Most of the math module’s functions are thin wrappers around the C platform’s mathematical functions. Since its underlying functions are written in CPython, the math module is efficient and conforms to the C standard.

The Python math module offers you the ability to perform common and useful mathematical calculations within your application. Here are a few practical uses for the math module:

* Calculating combinations and permutations using factorials
* Calculating the height of a pole using trigonometric functions
* Calculating radioactive decay using the exponential function
* Calculating the curve of a suspension bridge using hyperbolic functions
* Solving quadratic equations
* Simulating periodic functions, such as sound and light waves, using trigonometric functions

Since the math module comes packaged with the Python release, you don’t have to install it separately.

The Python math module offers a variety of predefined constants. Having access to these constants provides several advantages. For one, you don’t have to manually hard-code them into your application, which saves you a lot of time. Plus, they provide consistency throughout your code. The module includes several famous mathematical constants and important values:

* Pi
* Tau
* Euler’s number
* Infinity
* Not a number (NaN)

The Python math module provides functions that are useful in number theory as well as in representation theory, a related field. These functions allow you to calculate a range of important values, including the following:

* The factorials of a number
* The greatest common divisor of two numbers
* The sum of iterables

1)

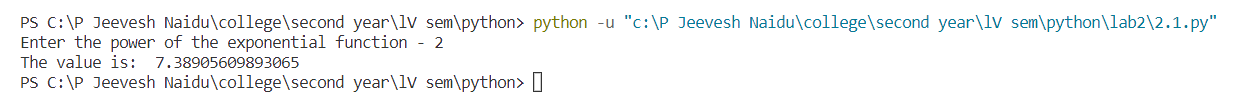
**Code:**

from math import exp

x = int(input("Enter the power of the exponential function."))

print("The value is: ", exp(x))

**Output:**

****

2)

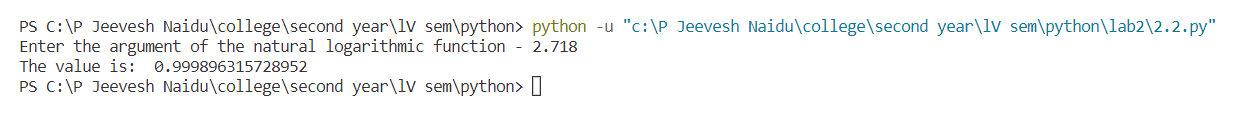
**Code:**

from math import log

x = float(input("Enter the argument of the natural logarithmic function."))

print("The value is: ", log(x))

**Output:**



3)

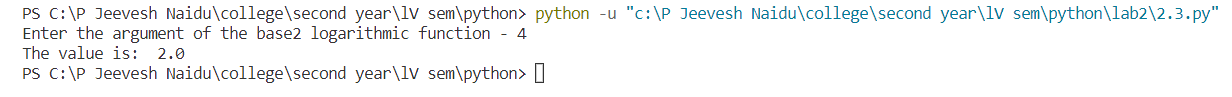
**Code:**

from math import log2

x = float(input("Enter the argument of the base2 logarithmic function."))

print("The value is: ", log2(x))

**Output:**

****

4)

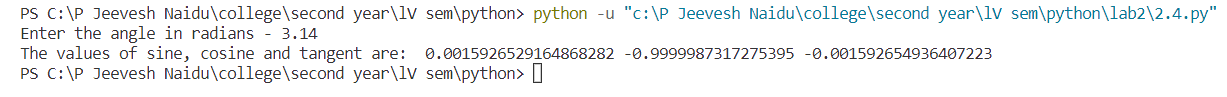
**Code:**

from math import sin, cos, tan

x = float(input("Enter the angle in radians."))

print("The values of sine, cosine and tangent are: ", sin(x), cos(x), tan(x))

**Output:**

****

5)

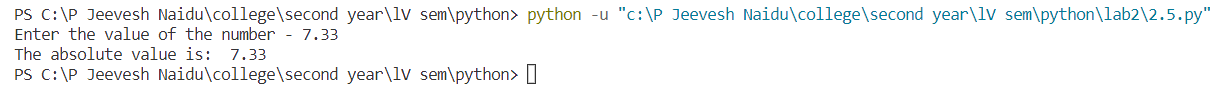
**Code:**

from math import fabs

x = float(input("Enter the value of the number."))

print("The absolute value is: ", fabs(x))

**Output:**

****

6)

**Code:**

def deg():

deg = float(input("Enter the measure in degrees."))

print("The radian measure is: ", deg\*3.14/180)

def rad():

rad = float(input("Enter the measure in radians."))

print("The degree measure is: ", rad\*180/3.14)

inp = int(input("Enter 1 for degrees and 2 for radians."))

if (inp == 1):

deg()

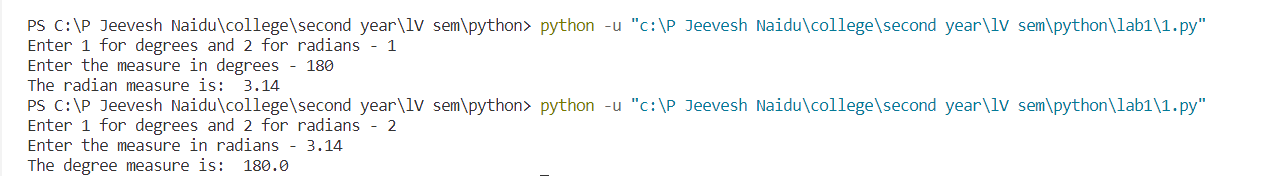
elif (inp == 2):

rad()

else:

print("Invalid input.")

**Output:**

****

7a)

**Code:**

from math import sin, cos, tan, isclose

n = float(input("Enter the value of angle."))

lhs = cos(n)/((1/cos(n))+tan(n))

rhs = 1-sin(n)

print(lhs)

print(rhs)

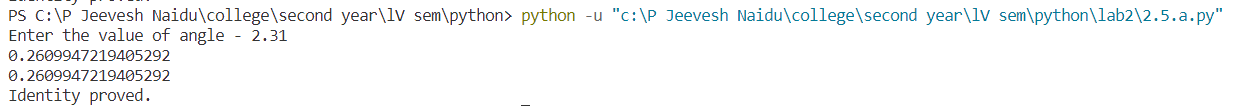
if (isclose(lhs, rhs)):

print("Identity proved.")

quit()

print("Identity failed to prove.")

**Output:**

****

7b)

**Code:**

from math import sin, tan, isclose

n = float(input("Enter the value of angle."))

lhs = (1/sin(n))+(1/tan(n))

rhs = (1/tan(n/2))

print(lhs)

print(rhs)

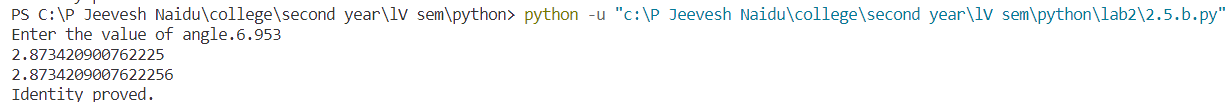
if (isclose(lhs, rhs)):

print("Identity proved.")

quit()

print("Identity failed to prove.")

**Output:**

****

7c)

**Code:**

from math import sin, cos, isclose

n = float(input("Enter the value of angle."))

lhs = ((1/cos(n)+1))/(1/cos(n)\*\*2)

rhs = (sin(n)\*\*2)/((1/cos(n))-1)

print(lhs)

print(rhs)

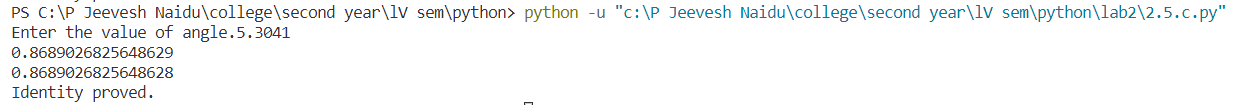
if (isclose(lhs, rhs)):

print("Identity proved.")

quit()

print("Identity failed to prove.")

**Output:**

****

7d)

**Code:**

from math import sin, cos, isclose, pi

n = float(input("Enter the value of angle."))

lhs = sin(pi/2-n)

rhs = cos(n)

print(lhs)

print(rhs)

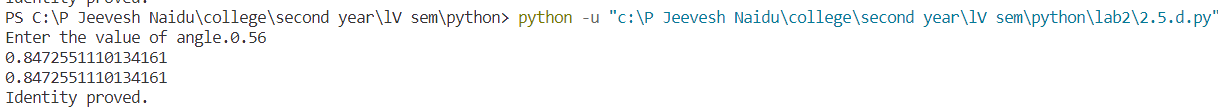
if (isclose(lhs, rhs)):

print("Identity proved.")

quit()

print("Identity failed to prove.")

**Output:**

****

7e)

**Code:**

from math import sin, cos, isclose, pi, sqrt

n = float(input("Enter the value of angle."))

lhs = cos(n)+sin(n)

rhs = sqrt(2)\*sin(n+pi/4)

print(lhs)

print(rhs)

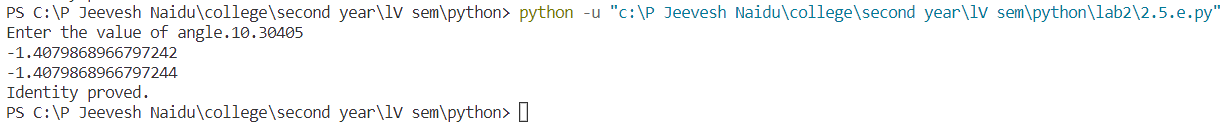
if (isclose(lhs, rhs)):

print("Identity proved.")

quit()

print("Identity failed to prove.")

**Output:**

****