1) Write a python program which creates a class named Cone and write a function calculate_area which calculates the area of the Cone.

In [34]:

```
import math
class Cone:
    def __init__(self):
        self.r = int(input("Enter the radius:"))
        self.h = int(input("Enter the hieght:"))
        self.calculate_area()

    def calculate_area(self):
        l = math.sqrt(self.r * self.r + self.h * self.h)
        area = math.pi * self.r * (self.r + l)
        print("Surface area of a cone: {}".format(area))
```

In [35]:

```
x = Cone()
```

```
Enter the radius:1
Enter the hieght:1
Surface area of a cone: 7.584475591748159
```

2) Define a class MathOperation which implements pow(x,n) without using python's in-built pow() method

Sample Execution:

```
M = MathOperation()
print(M.pow(2, 3))
8
print(M.pow(5, -3))
0.008
print(M.pow(-2, 5))
-32
print(M.pow(-5, -3))
-0.008
print(M.pow(20000,0))
1
```

In [54]:

```
class MathOpertaion:
    def __init__(self):
        pass
    def power(self,a,b):
        return a**b
```

In [55]:

```
M=MathOpertaion()
```

In [56]:

```
print(M.power(2,3))
```

```
8
```

In [58]:

```
print(M.power(5, -3))
```

```
0.008
```

In [59]:

```
print(M.power(-2, 5))
```

```
-32
```

3) Write a python program that creates a class Base and Derived. Use inbuilt function issubclass and isinstance which gives boolean results.(True or False)

Check:
Derived class is a subclass of Base class which will return true
Base class is a subclass of Derived class which will return false
Base class is an instance of Derived class which will return false
Derived class is an instance of Base class which will return true

In [61]:

```
class Base:
    def __init__(self):
        pass
```

In [63]:

```
class Derived(Base):
    def __init__(self):
        pass
```

In [71]:

```
issubclass(Base,Derived)
```

Out[71]:

```
False
```

In [72]:

```
issubclass(Derived,Base)
```

Out[72]:

True

In [73]:

```
isinstance(Base,Derived)
```

Out[73]:

False

In [74]:

```
isinstance(Derived,Base)
```

Out[74]:

False

4) Write a python program that creates base class Person which has two methods

```
def __init__(self, first, last)
def __str__(self)
Also create a derived class named Employee which uses the base class
method “def __str__(self)” using “super()” to concatenate first name with last nam
e
```

In [22]:

```
class Person:
    def __init__(self,first,last):
        self.first = first
        self.last = last

    def __str__(self):
        pass
```

In [25]:

```
class Employee(Person):
    def __init__(self,first,last):
        super().__init__(first,last)
        self.__str__()

    def __str__(self):
        print(self.first + " " + self.last)
```

In [27]:

```
e = Employee("Sudipta","Das")
```

```
Sudipta Das
```