

## Experiment - 11

**Aim:** Implementation of Data Abstraction in Python.

**Theory:**

- In Python, abstraction can be achieved by using abstract classes.
- A class that consists of one or more abstract method is called the abstract class.
- Abstract methods do not contain their implementation.
- Abstract class can be inherited by the subclass and abstract methods get their definition in the subclass.
- Python provides the abc (abstract base class) module with inbuilt ABC class to use the abstraction.

**Program:**

```
# Python program to define abstract class
from abc import ABC
```

```
class Polygon(ABC):
    #abstract method
    def sides(self):
        pass

    #normal method
    def a(self):
        print("\nThis is general method.\n")
```

```
class Triangle(Polygon):
    def sides(self):
        print("Triangle has 3 sides")
```

```
class Pentagon(Polygon):
    def sides(self):
        print("Pentagon has 5 sides")
```

```
class Hexagon(Polygon):
    def sides(self):
        print("Hexagon has 6 sides")
```

```
class square(Polygon):
    def sides(self):
        print("Square has 4 sides")
```

**#Object of Child Class**

```
t = Triangle()
t.sides()
t.a()
```

```
s = square()
s.sides()
```

```
p = Pentagon()
p.sides()
```

```
k = Hexagon()
k.sides()
```

**Output:**

```
Triangle has 3 sides
This is general method.
Square has 4 sides
Pentagon has 5 sides
Hexagon has 6 sides
```

**Conclusion:** Hence, implemented the concept of Data Abstraction.