```
from abc import ABC, abstractmethod

# Abstract Class need to implemented
class Parent(ABC):

   @abstractmethod
   def d1(self):
     pass

p = Parent()
p.d1()
```

```
TypeError: Can't instantiate abstract class Parent with abstract method d1
```

```
from abc import ABC, abstractmethod

# Abstract Class need to implemented in Child Class
class Parent(ABC):

   @abstractmethod
   def d1(self):
     pass

class Child(Parent):

   def d1(self):
      print("d1 method implemented")

c = Child()
c.d1()
```

```
d1 method implemented
```

```
from abc import ABC, abstractmethod

# All methods in Abstract Class
class Parent(ABC): # Is subclassed by: Child

    @abstractmethod
    def d1(self): # Is overridden in: Child
      pass

    @classmethod
    @abstractmethod
    def d2(cls): # Is overridden in: Child
       pass

    @staticmethod
    @abstractmethod
    def d3(): # Is overridden in: Child
       pass

class Child(Parent):

    def d1(self): # Overrides method in Parent
       print("d1 method implemented")

    def d2(self): # Overrides method in Parent
       print("d2 method implemented")

    def d3(self): # Overrides method in Parent
       print("d3 method implemented")

c = Child()
c.d1()
c.d2()
c.d3()
```

```
# Constructors in Abstract Class
from abc import ABC, abstractmethod

class Parent(ABC):

   @abstractmethod
   def __init__(self):
      pass

Parent()

TypeError: Can't instantiate abstract class Parent with abstract method __init__
```

```
# Constructors in Abstract Class implemented
from abc import ABC, abstractmethod

class Parent(ABC): # Is subclassed by:   Child

   @abstractmethod
   def __init__(self):
      pass

class Child(Parent): # Is overridden in:   Child

   def __init__(self):
      print("Special Method")

Child()

Special Method
```

```
from abc import ABC, abstractmethod

# When Abstract Class contains all abstract methods known as Interface
class Parent(ABC): # Is subclassed by:   Child

    @abstractmethod
    def d1(self): # Is overridden in:   Child
      pass

    @abstractmethod
    def d2(self): # Is overridden in:   Child
       pass

class Child(Parent):

    def d1(self): # Overrides method in Parent
       print("d1 method implemented")

    def d2(self): # Overrides method in Parent
       print("d2 method implemented")

c = Child()
c.d1()
c.d2()
```

```
d1 method implemented
d2 method implemented
```