
9. Abstract Class

1. Create an abstract class with abstract and non-abstract methods.
2. Create a sub class for an abstract class. Create an object in the child class for the abstract class and access the non-abstract methods
3. Create an instance for the child class in child class and call abstract methods
4. Create an instance for the child class in child class and call non-abstract methods



main.py

Shell



```
1  # 1st program...
2  from abc import ABC, abstractmethod
3  class Shape(ABC):
4      @abstractmethod
5      def area(self):
6          pass
7      def perimeter(self):
8          return 0
9  class Square(Shape):
10     def __init__(self, side):
11         self.side = side
12     def area(self):
13         return self.side * self.side
14  class Circle(Shape):
15     def __init__(self, radius):
16         self.radius = radius
17     def area(self):
18         return 3.14 * self.radius ** 2
19  square = Square(5)
20  print(square.area())
21  circle = Circle(3)
22  print(circle.area())
23
```



main.py

Shell



25

28.26

> |



main.py

Shell



```
1  # 2nd program...
2  from abc import ABC, abstractmethod
3
4  class Animal(ABC):
5      @abstractmethod
6      def speak(self):
7          pass
8
9  class Dog(Animal):
10     def speak(self):
11         return "Woof!"
12
13  d = Dog()
14  print(d.speak())
15  |
```



main.py

Shell



Woof!

> |



main.py

Shell



```
2 from abc import ABC, abstractmethod
3 class Shape(ABC):
4     @abstractmethod
5     def area(self):
6         pass
7     def perimeter(self):
8         return 0
9 class Square(Shape):
10     def __init__(self, side):
11         self.side = side
12     def area(self):
13         return self.side * self.side
14     def draw(self):
15         print("Drawing a square with
            side", self.side)
16 class Circle(Shape):
17     def __init__(self, radius):
18         self.radius = radius
19     def area(self):
20         return 3.14 * self.radius ** 2
21     def draw(self):
22         print("Drawing a circle with
            radius", self.radius)
23 square = Square(5)
24 print(square.area())
25 square.draw()
26 circle = Circle(3)
27 print(circle.area())
28 circle.draw()
```



main.py

Shell



25

Drawing a square with side 5

28.26

Drawing a circle with radius 3

>



main.py

Shell



```
1  #4th program...
2  class Parent:
3      def __init__(self, name):
4          self.name = name
5
6      def say_hello(self):
7          print("Hello, my name is", self
              .name)
8
9  class Child(Parent):
10     def __init__(self, name, age):
11         super().__init__(name)
12         self.age = age
13
14     def say_age(self):
15         print("I am", self.age, "years
              old")
16
17
18  child = Child("Santhosh kumar", 23)
19
20  child.say_hello()
21  child.say_age()
```




main.py

Shell



Hello, my name is Santhosh kumar

I am 23 years old

> |