```python
from abc import ABC, abstractmethod

# Abstract Base Class defining a common interface
class Shape(ABC):
    @abstractmethod
    def area(self):
        pass

    @abstractmethod
    def perimeter(self):
        pass

# Child class 1: Circle
class Circle(Shape):
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        return 3.14 * self.radius ** 2

    def perimeter(self):
        return 2 * 3.14 * self.radius

# Child class 2: Rectangle
class Rectangle(Shape):
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def area(self):
        return self.length * self.width

    def perimeter(self):
        return 2 * (self.length + self.width)

# Child class 3: Square
class Square(Rectangle):
    def __init__(self, side_length):
        # Square is a special case of Rectangle where length == width
        super().__init__(side_length, side_length)

# Usage example
circle = Circle(radius=5)
rectangle = Rectangle(length=4, width=6)
square = Square(side_length=3)

print(f"Circle Area: {circle.area()}, Perimeter: {circle.perimeter()}")
print(f"Rectangle Area: {rectangle.area()}, Perimeter: {rectangle.perimeter()}")
print(f"Square Area: {square.area()}, Perimeter: {square.perimeter()}")
```

```
Log: User created
Saving to database: {'user_id': 1, 'name': 'John'}
```