SOLID PRINCIPLES

10/10

Single ✓

A class should have only one reason to change. meaning it should have only one job or responsibility.
This makes the class easier to understand and maintain

```
class User
  def __init__(self, name, email):
    self.name = name
    self.email = email
  def send_email(self, message):
    print(f"Sending email to {user.email}: {message}")
user = User("John Clive", "john@example.com")
email_service = EmailService()
email_service.send_email(user, "Hello, John!")
```

By ensuring that a class has only one responsibility, we can easily change or refactor that part of the code without other parts

Open / Closed ✓

Software entities (classes, modules, functions, etc) should be open for extension but closed for modification. This means you can add new functionality without changing code.

```
class Rectangle
  def draw(self):
    print("Drawing a rectangle")
class Circle:
  def draw(self):
    print("Drawing a circle")
def draw_shape(shape):
  if isinstance(shape, Rectangle):
    shape.draw()
  elif isinstance(shape, Circle):
    shape.draw()
```

We protect existing code from being modified when new features are added, which reduces the risk of introducing bugs.