```
# A simple representation of a User class for demonstration purposes.
# In a real application, this would likely be a database model.
class User:
    def __init__(self, username, is_registered=False,
is_paid_member=False):
        self.username = username
        self.is_registered = is_registered
        self.is_paid_member = is_paid_member

    def __repr__(self):
        return f"User(username='{self.username}',
is_registered={self.is_registered},
is_paid_member={self.is_paid_member})"

# A decorator to check if a user is registered.
def check_registered(func):
    """
    This decorator ensures that the decorated function is only
executed
    if the user object passed to it is a registered member.
    """
    def wrapper(user, *args, **kwargs):
        if user.is_registered:
            return func(user, *args, **kwargs)
        else:
            print(f"Access Denied: {user.username} is not a registered
user.")
    return wrapper

# A decorator to check if a user is a paid member.
# This decorator is typically applied after the registration check.
def check_paid_member(func):
    """
    This decorator ensures that the decorated function is only
executed
    if the user object is a paid member.
    """
    def wrapper(user, *args, **kwargs):
        if user.is_paid_member:
            return func(user, *args, **kwargs)
        else:
            print(f"Access Denied: {user.username} is not a paid
member.")
    return wrapper

# Functions representing blog features, decorated with our
authentication logic.
```

```
@check_registered
def read_blog_post(user, post_id):
    """Allows a registered user to read a blog post."""
    print(f"{user.username} is reading blog post {post_id}.")

@check_registered
def write_comment(user, post_id, comment_text):
    """Allows a registered user to write a comment."""
    print(f"{user.username} is writing a comment on post {post_id}:
'{comment_text}'")

@check_registered
@check_paid_member
def access_premium_content(user, content_id):
    """Allows a paid and registered user to access premium content."""
    print(f"{user.username} is accessing premium content
{content_id}.")

# --- Main execution to demonstrate the system ---

if __name__ == "__main__":
    # Create sample users
    guest_user = User("guest_user")
    free_member = User("free_member", is_registered=True)
    premium_member = User("premium_member", is_registered=True,
is_paid_member=True)

    print("--- Attempting actions for guest_user (not registered)
---")
    read_blog_post(guest_user, 1)
    write_comment(guest_user, 1, "This is a comment.")
    access_premium_content(guest_user, "premium_video_1")
    print("-" * 40)

    print("--- Attempting actions for free_member (registered, not
paid) ---")
    read_blog_post(free_member, 2)
    write_comment(free_member, 2, "I love this blog!")
    access_premium_content(free_member, "premium_video_2")
    print("-" * 40)

    print("--- Attempting actions for premium_member (registered and
paid) ---")
    read_blog_post(premium_member, 3)
    write_comment(premium_member, 3, "Great content, thanks!")
    access_premium_content(premium_member, "premium_video_3")
    print("-" * 40)
```