```python
def perform_gradient_descent(initial_x, initial_y, objective_function, gradient_function, learning_rate, iterations):
    x_val = initial_x
    y_val = initial_y

    # Iterate for Gradient Descent
    for i in range(iterations):
        gradient_x, gradient_y = gradient_function(x_val, y_val)

        # Update x and y using Gradient Descent
        x_val = x_val - learning_rate * gradient_x
        y_val = y_val - learning_rate * gradient_y

    return x_val, y_val

def objective_function(x, y):
    return x**2 + y**2

def compute_gradient(x, y):
    grad_x = 2 * x
    grad_y = 2 * y
    return grad_x, grad_y

# Parameters as given in the provided image
initial_x = 0.1
initial_y = 0.1
learning_rate = 0.1
iterations = 10

# Calling the Gradient Descent Function
optimal_solution = perform_gradient_descent(initial_x, initial_y, objective_function, compute_gradient, learning_rate, iterations)
optimal_solution
```

```
[1]: (0.010737418240000003, 0.010737418240000003)
```