In [1]:

```
import matplotlib.pyplot as plt
```

In [2]:

```
def f(x):
 return (x + 3)**2
def df(x):
 return 2 * (x + 3)
```

In [3]:

```
x_old = 2 # Starting point
learning_rate = 0.1
precision = 0.00001
max_iterations = 1000
```

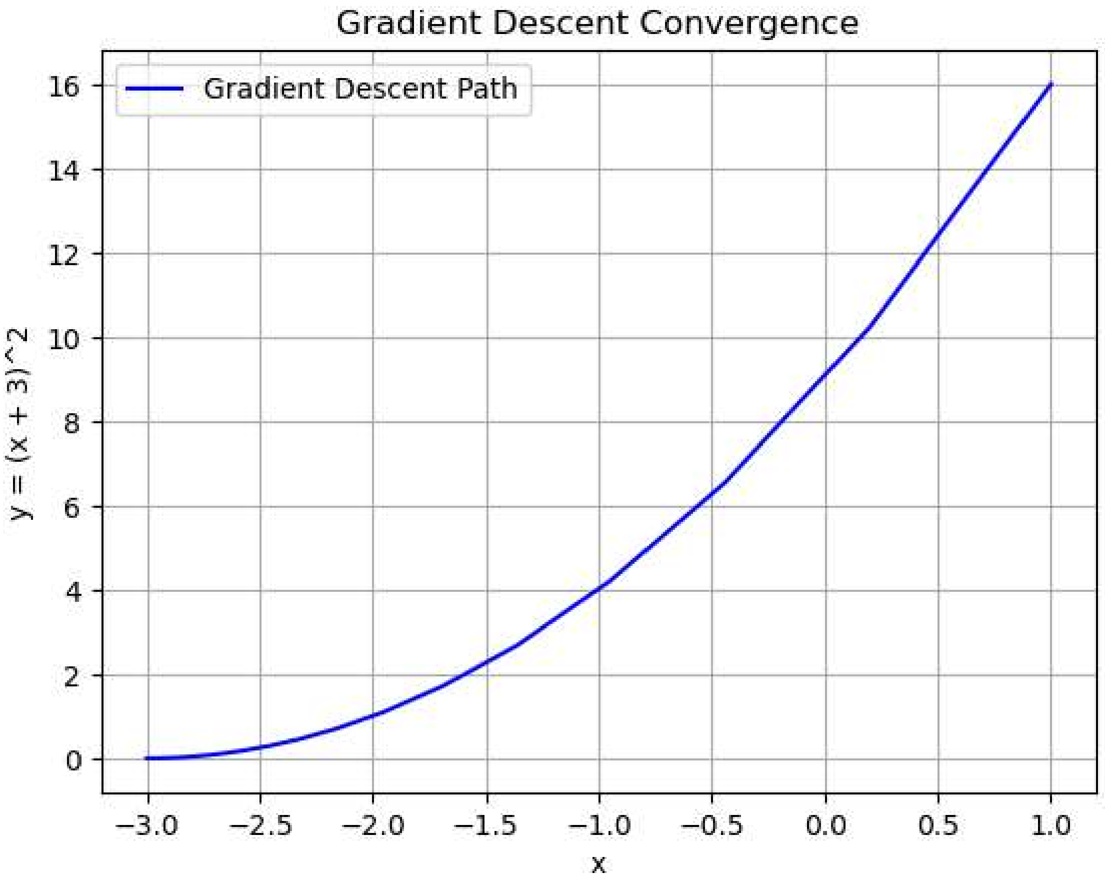
In [5]:

```
x_values = []
y_values = []
for i in range(max_iterations):
    gradient = df(x_old)
    x_new = x_old - learning_rate * gradient
    x_values.append(x_new)
    y_values.append(f(x_new))
    if abs(x_new - x_old) < precision:
         break
    x_old = x_new
print(f"Local minima occurs at x = {x_new:.5f}")
print(f"Minimum value of function y = {f(x_new):.5f}")
```

```
Local minima occurs at x = -2.99996
Minimum value of function y = 0.00000
```

In [11]:

```
plt.plot(x_values, y_values, 'b-', label='Gradient Descent Path')
plt.title("Gradient Descent Convergence")
plt.xlabel("x")
plt.ylabel("y = (x + 3)^2")
plt.legend()
plt.grid(True)
plt.show()
```


Gradient Descent Convergence
Gradient Descent Path
y = (x + 3)^2
x
16
14
12
10
8
6
4
2
0
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0