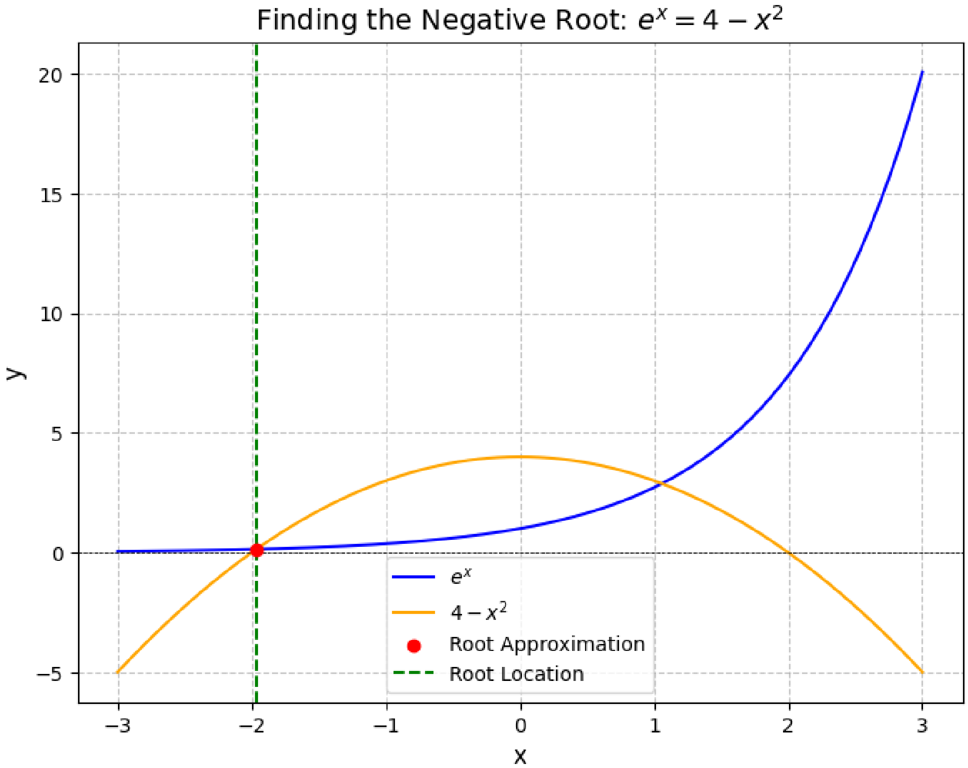
```
import numpy as np
import matplotlib.pyplot as plt

# Function and derivative definitions
def function(x):
    # Defines the equation: f(x) = e^x - (4 - x^2)
    return np.exp(x) - (4 - x**2)

def derivative(x):
    # Defines the derivative of f(x): f'(x) = e^x + 2x
    return np.exp(x) + 2*x

# Initial guess and tolerance for Newton's method
initial_guess = -1  # Starting point for Newton's method
tolerance = 1e-6  # Stopping criterion for iteration
max_iterations = 100  # Maximum allowed iterations

# Newton's Method implementation
x_current = initial_guess  # Initialize current approximation
for iteration in range(max_iterations):
    # Compute the next approximation using Newton's formula
    x_new = x_current - function(x_current) / derivative(x_current)
    # Check if the difference between consecutive approximations is within tolerance
    if abs(x_new - x_current) < tolerance:
        break
    # Update the current approximation
    x_current = x_new

# Generate data for the plot
x_range = np.linspace(-3, 3, 400)  # Create a range of x values for plotting
y_exp = np.exp(x_range)  # Calculate y-values for the exponential function
y_parabola = 4 - x_range**2  # Calculate y-values for the parabola function

# Plotting the functions and the approximated root
plt.figure(figsize=(8, 6))  # Set the figure size
plt.plot(x_range, y_exp, label=r'$e^x$', color='blue')  # Plot the exponential function
plt.plot(x_range, y_parabola, label=r'$4 - x^2$', color='orange')  # Plot the parabola
plt.scatter(x_current, np.exp(x_current), color='red', label='Root Approximation', zorder=5)  # Mark the approximated roo

# Adding labels and details
plt.axhline(0, color='black', linewidth=0.5, linestyle='--')  # Add a horizontal line at y=0
plt.axvline(x=x_current, color='green', linestyle='--', label='Root Location')  # Add a vertical line at the approximatec
plt.title('Finding the Negative Root: $e^x = 4 - x^2$', fontsize=14)  # Title of the plot
plt.xlabel('x', fontsize=12)  # Label for x-axis
plt.ylabel('y', fontsize=12)  # Label for y-axis
plt.grid(True, linestyle='--', alpha=0.7)  # Add a grid with dashed lines
plt.legend(fontsize=10)  # Add a legend with specified font size

# Display the plot
plt.show()  # Render the plot
```


Finding the Negative Root: $e^x = 4 - x^2$
20
15
10
5
0
−5
y
−3
−2
−1
0
1
2
3
x
e^x
$4 - x^2$
Root Approximation
Root Location