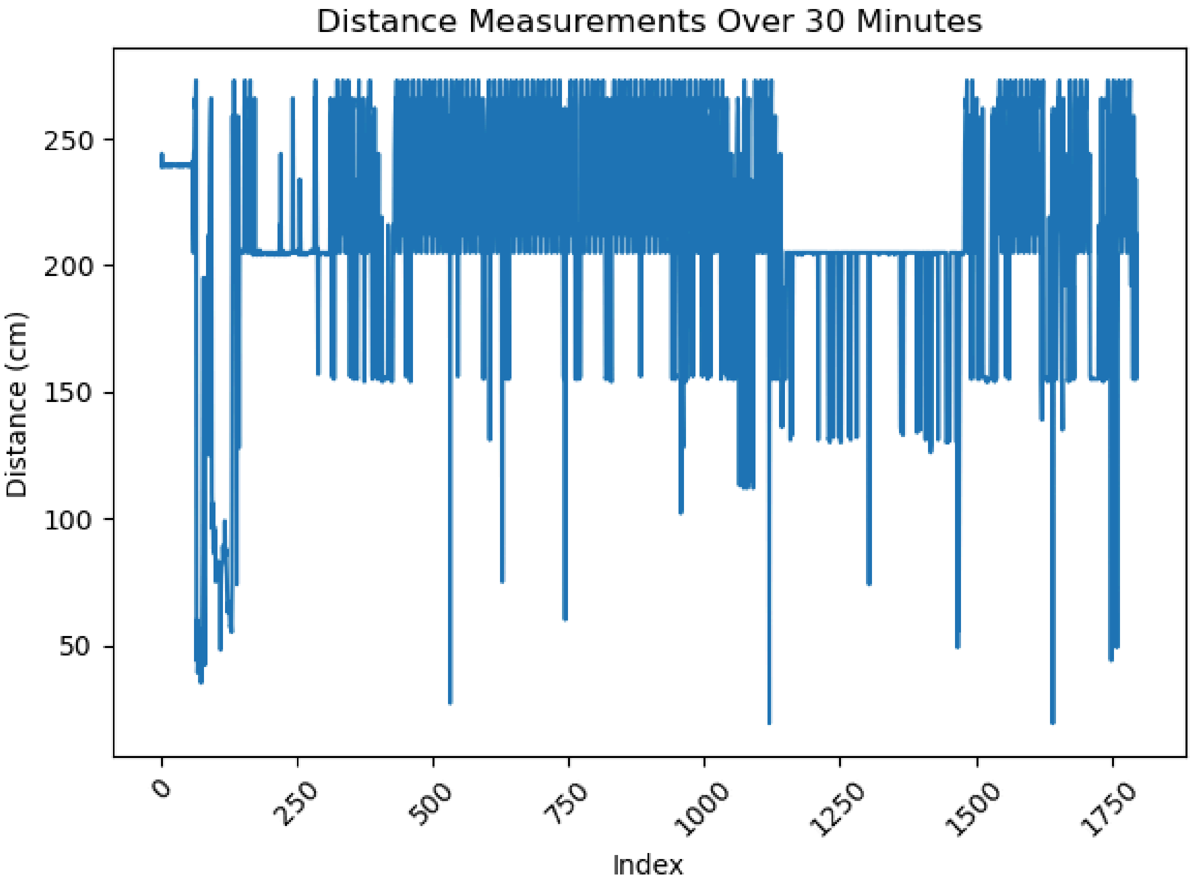
In [1]:

```
student_name = "Rammaka Aaron Iddamalgoda"  # fill your name
student_id = "s223496576"  # fill your student ID
print("Student name: " + student_name)
print("Student ID: " + student_id)
```

```
Student name: Rammaka Aaron Iddamalgoda
Student ID: s223496576
```

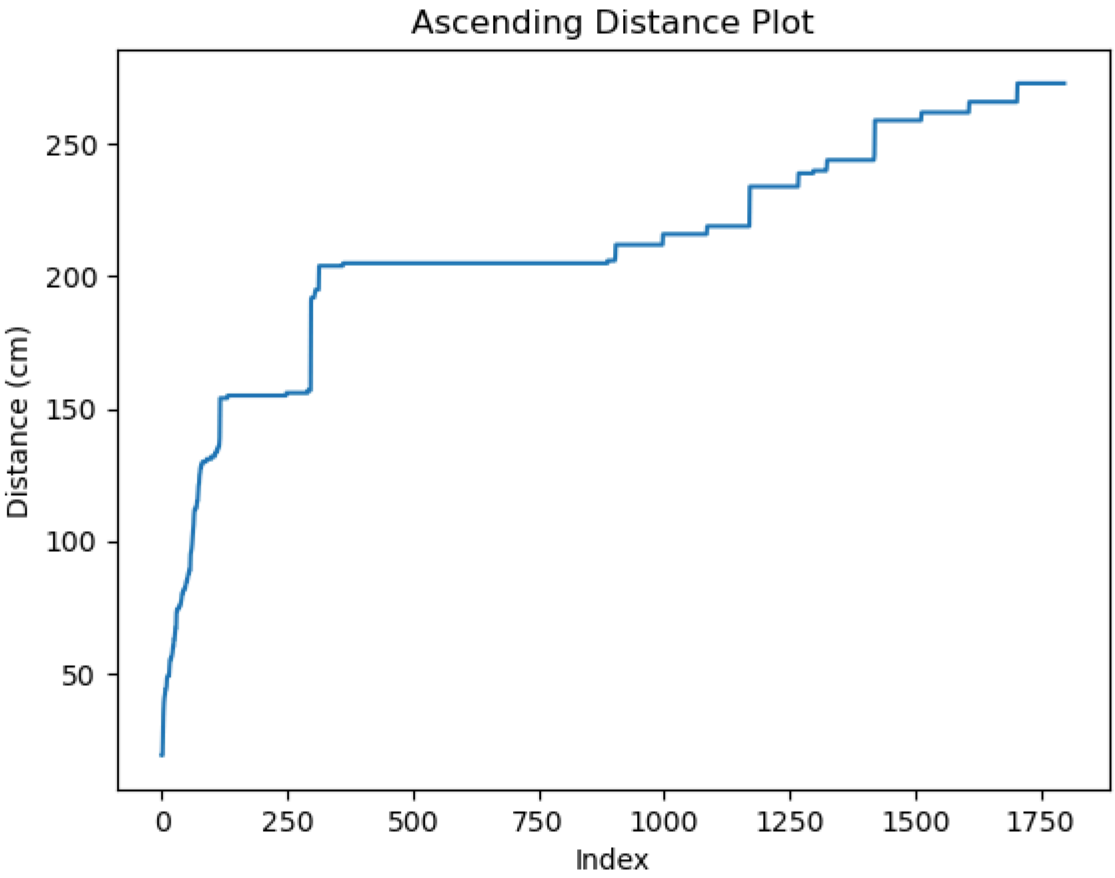
In [6]:

```
import math
import random
import pandas as pd
import matplotlib.pyplot as plt

# Read the CSV file
df = pd.read_csv('/PythonPrograms/distance_data1.csv')

# Use the 'Timestamp' column for the x-axis and 'Distance' for the y-axis
# You can convert the Timestamp to a more readable format if needed
df['Timestamp'] = pd.to_datetime(df['Timestamp'], format='%Y%m%d%H%M%S')

# Alternatively, use the index for the x-axis (0 to 1799 for 1800 data point
x_values = range(len(df))

# Plot distance over time (Timestamp or index)
plt.plot(x_values, df['Distance'])
plt.xlabel('Index')
plt.ylabel('Distance (cm)')
plt.title('Distance Measurements Over 30 Minutes')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability if us
plt.tight_layout()  # Adjust layout to prevent labels from being cut off
plt.show()
```



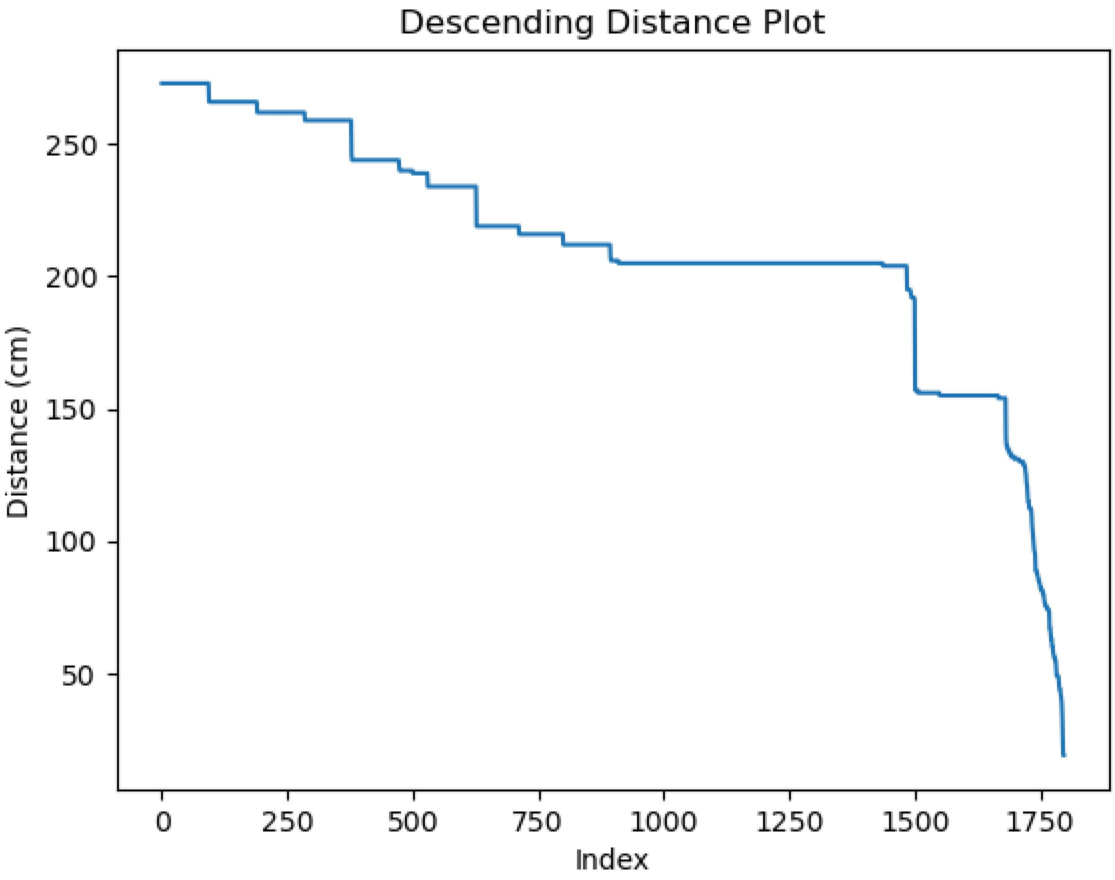
In [11]:

```
n_values = len(df)  # Number of data points (1800 for 30 minutes)
x_values = range(n_values)

# Activity 1: Sort the distance values in ascending order and plot
y_values_ascending = sorted(df['Distance'])  # Ascending values from CSV

plt.plot(x_values, y_values_ascending)
plt.xlabel('Index')
plt.ylabel('Distance (cm)')
plt.title('Ascending Distance Plot')
plt.show()
```



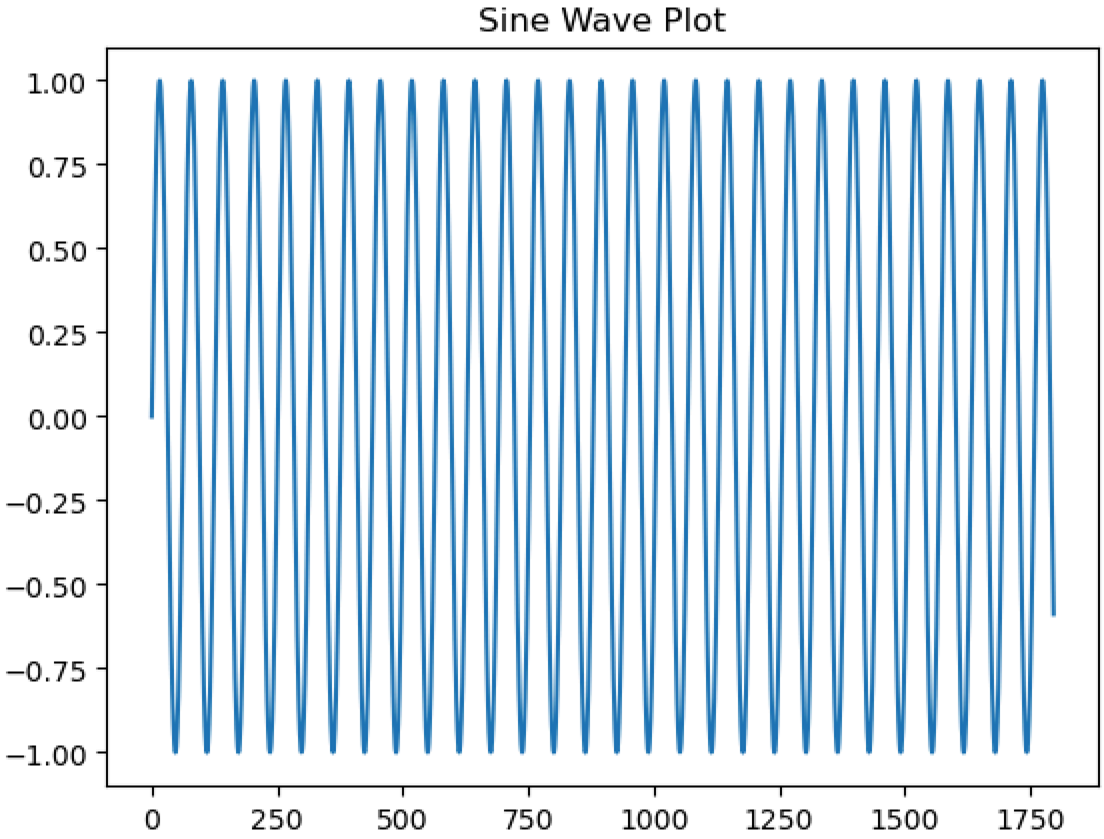
In [12]:

```
#
# Activity 2: Create data so that the plot draws a
# descending line (y_values decrease at any rate).
#

y_values_descending = sorted(df['Distance'], reverse=True)
plt.plot(x_values, y_values_descending)
plt.xlabel('Index')
plt.ylabel('Distance (cm)')
plt.title('Descending Distance Plot')
plt.show()
```



In [14]:

```
#
# Activity 3: Create data so that the plot draws a
# wave. You can consider using Python's math libarary, which has
# a sin function (detail https://www.w3schools.com/python/ref_math_sin.asp).
#

y_values_wave = [math.sin(i * 0.1) for i in range(n_values)]

plt.plot(x_values, y_values_wave)
plt.title('Sine Wave Plot')
plt.show()
```


Sine Wave Plot
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
250
500
750
1000
1250
1500
1750

```
In [ ]:
```