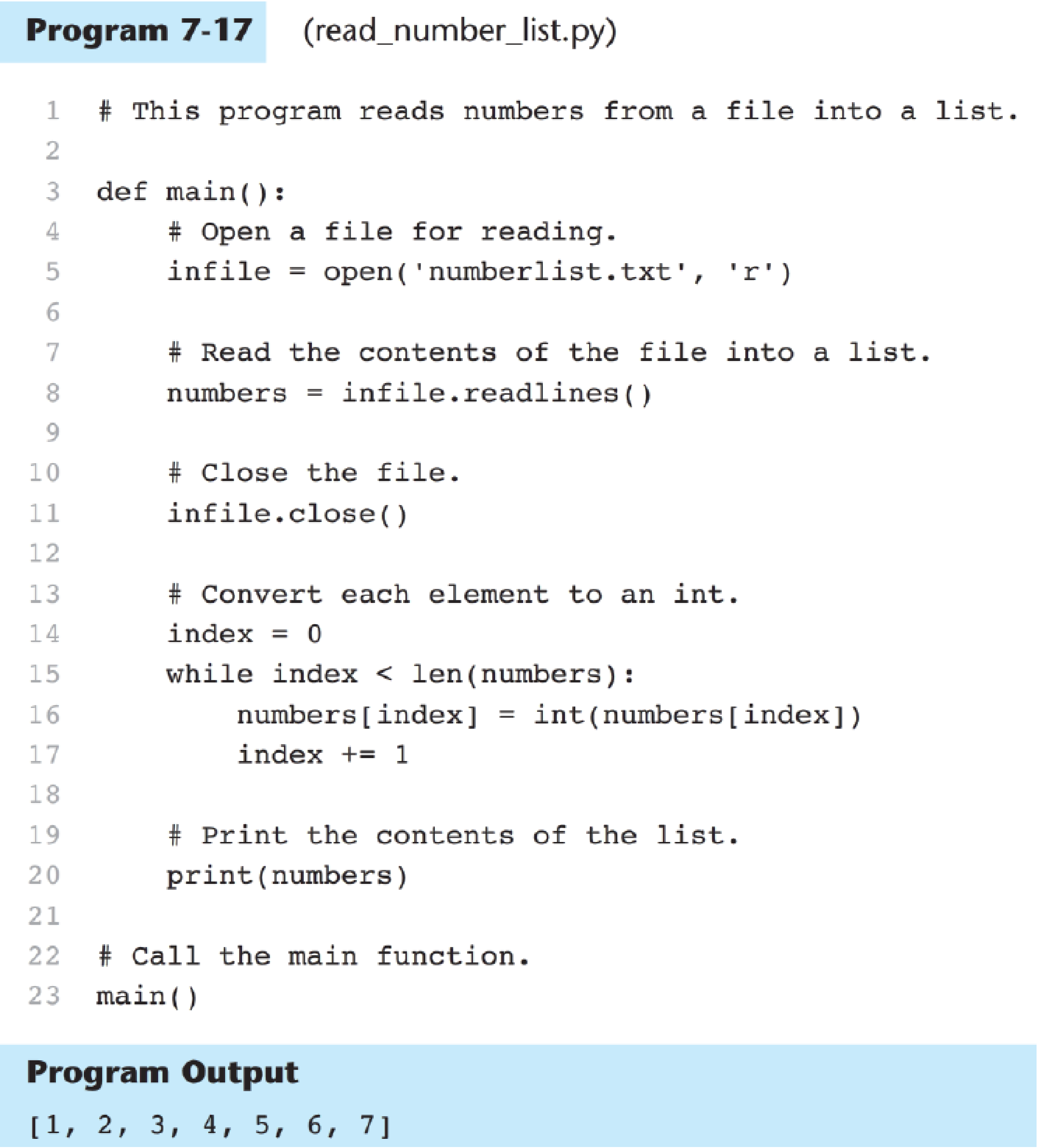
Laboratory 7
Making Graphs of Lists

COMSC-122

Fall 2017

Helpful Programs

- One of the things that you will need to do is to download files of numerical data, and transform them into lists.
- The following program should be helpful in this regard.



Program 7-17 (read_number_list.py)

```python
# This program reads numbers from a file into a list.

def main():
    # Open a file for reading.
    infile = open('numberlist.txt', 'r')

    # Read the contents of the file into a list.
    numbers = infile.readlines()

    # Close the file.
    infile.close()

    # Convert each element to an int.
    index = 0
    while index < len(numbers):
        numbers[index] = int(numbers[index])
        index += 1

    # Print the contents of the list.
    print(numbers)

# Call the main function.
main()
```

Program Output

```
[1, 2, 3, 4, 5, 6, 7]
```

Sample Program Useful for this Lab

Program 7-17:

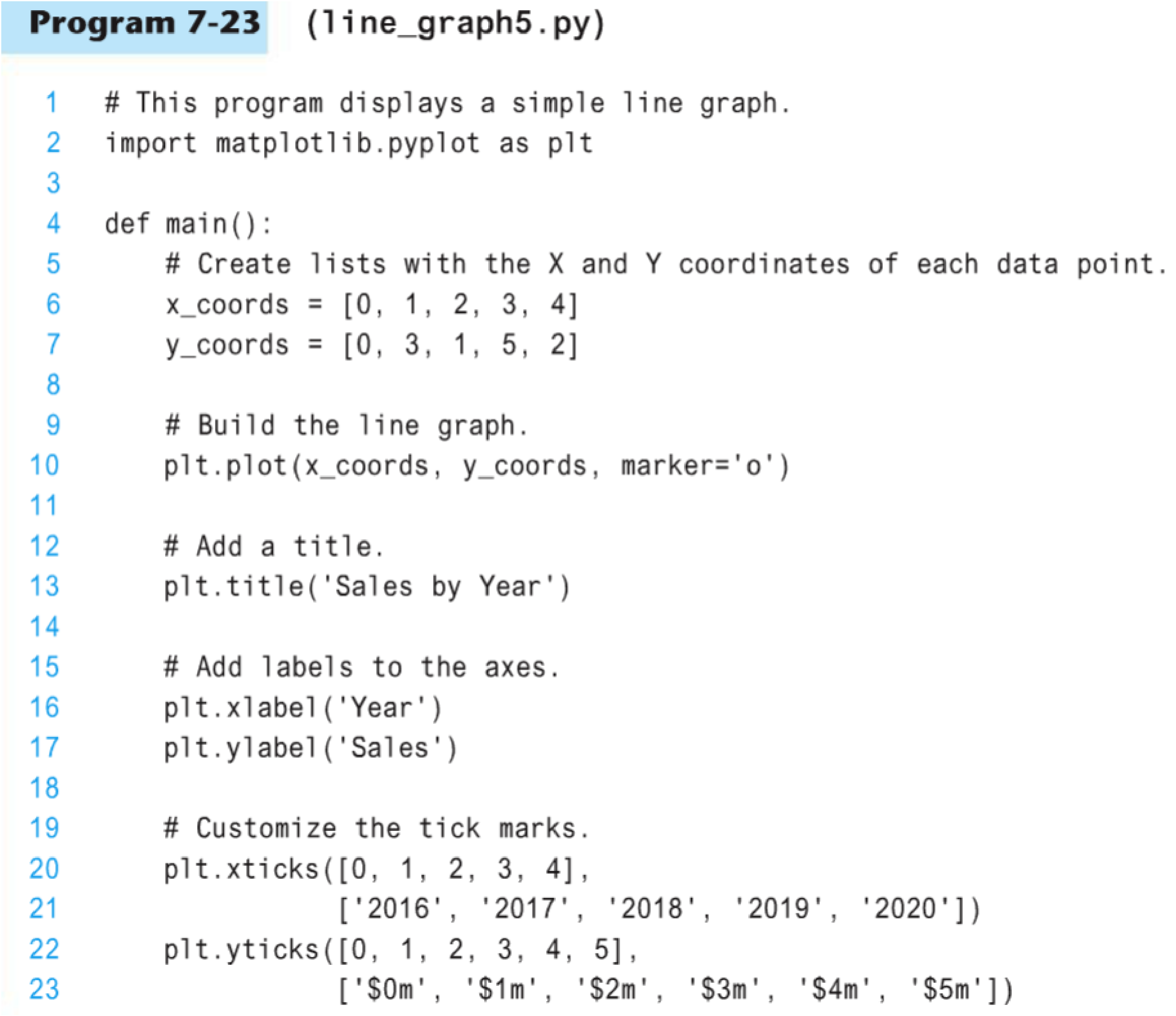
Reading a List of Numbers from a File

Lab07A Problem Statement

- Attached you will find a text file called 1994_Weekly_Gas_Average.txt .
- The file contains the average gas price for each week in the year 1994.
    - There are 52 lines in the file.
- Using matplotlib, write a Python program that reads the contents of the file, then plots the data as a line graph.
- Be sure to display meaningful labels along the X and Y axes.
- You may use the default tick marks in your display.
- Call the instructor over to review your work so that you may obtain proper credit.

Lab07A Pseudocode

- Prepare the **gas** list.
  - Open the text file 1994_Weekly_Gas_Average.txt .
  - Read the contents into a list called **gas** .
  - Close the file.
  - Strip the newline from each element.
- Prepare the **x_coords** list.
  - Create a list with 52 elements, with the numbers [1, 2, 3, …, 52]
  - Name the list **x_coords .**
- Plot the line graph.
  - Build the line graph
  - Limit the X axis to the range 1-52.
  - Add a Title (e.g. “1994 Weekly Gas Prices”)
  - Add labels to the X axis
  - Add labels to the Y axis
  - Display the graph.

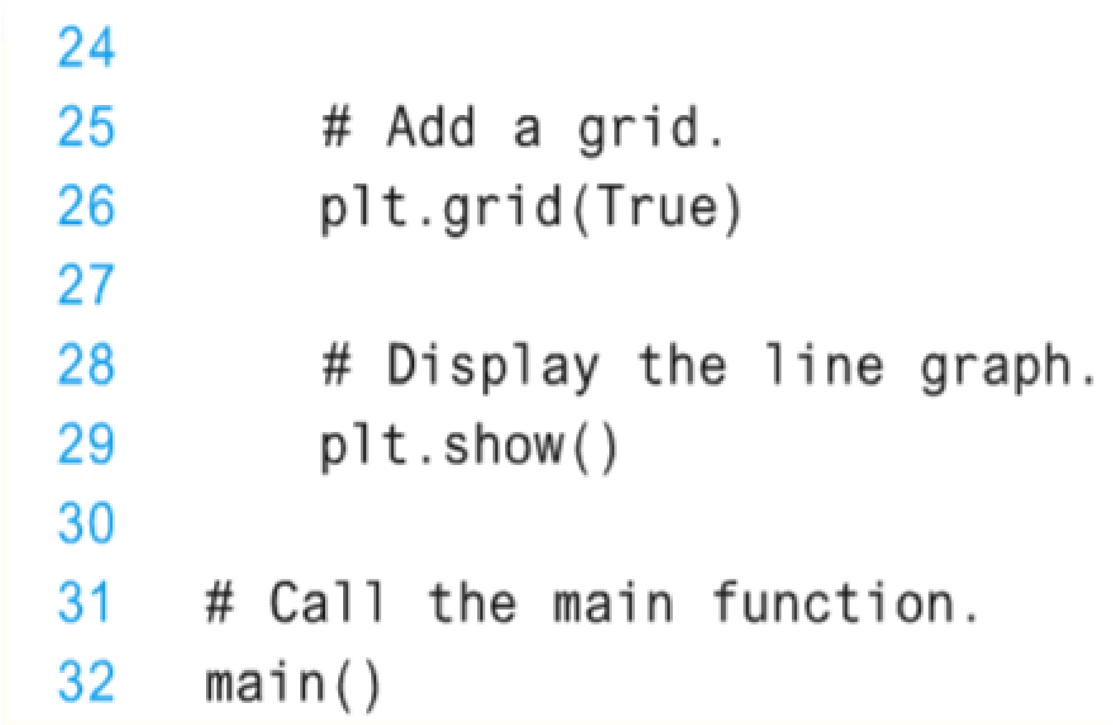


Program 7-23 (line_graph5.py)

```python
# This program displays a simple line graph.
import matplotlib.pyplot as plt

def main():
    # Create lists with the X and Y coordinates of each data point.
    x_coords = [0, 1, 2, 3, 4]
    y_coords = [0, 3, 1, 5, 2]

    # Build the line graph.
    plt.plot(x_coords, y_coords, marker='o')

    # Add a title.
    plt.title('Sales by Year')

    # Add labels to the axes.
    plt.xlabel('Year')
    plt.ylabel('Sales')

    # Customize the tick marks.
    plt.xticks([0, 1, 2, 3, 4],
               ['2016', '2017', '2018', '2019', '2020'])
    plt.yticks([0, 1, 2, 3, 4, 5],
               ['$0m', '$1m', '$2m', '$3m', '$4m', '$5m'])
```

Sample for Lab07A

Program 7-23
line_graph5.py
part 1

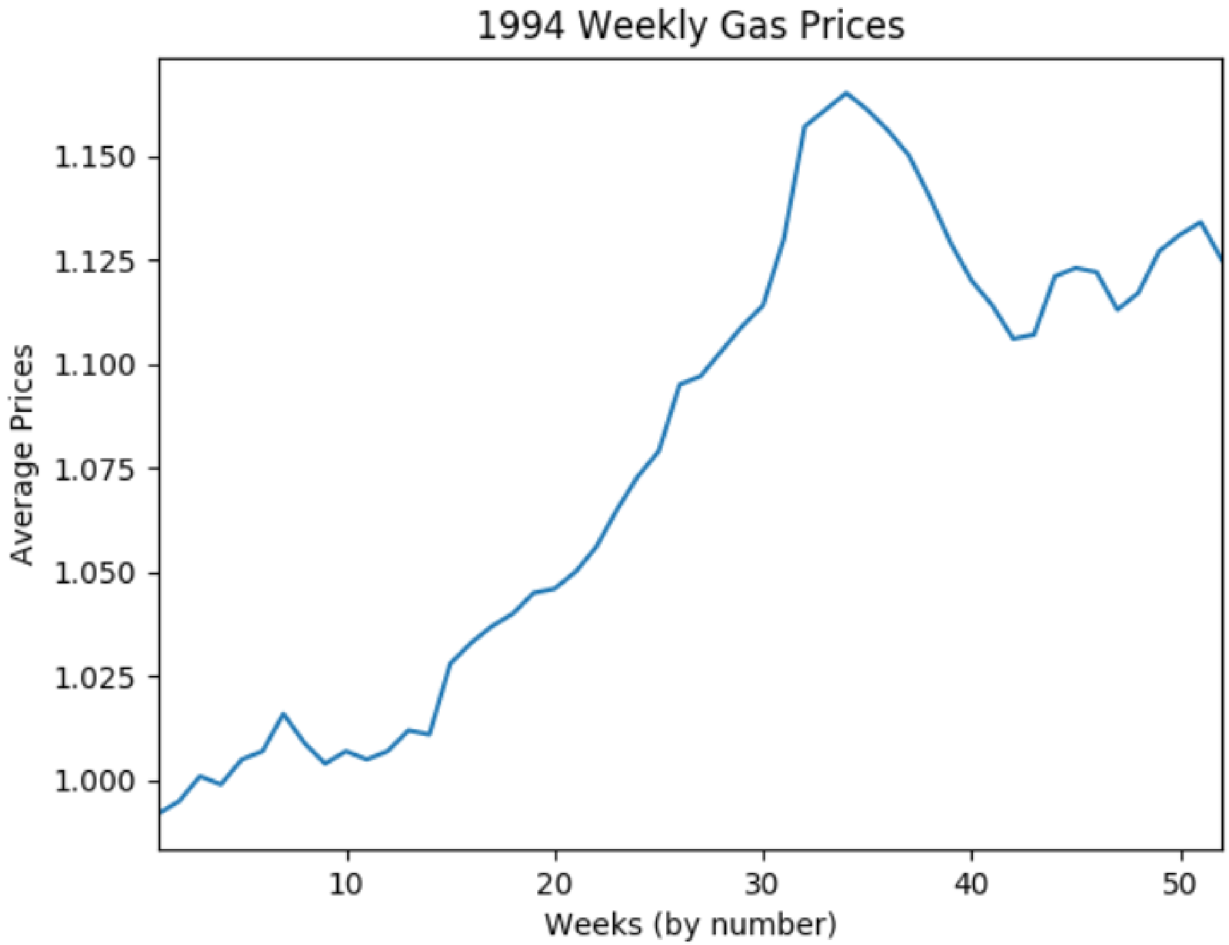


```python

    # Add a grid.
    plt.grid(True)

    # Display the line graph.
    plt.show()

# Call the main function.
main()
```

Sample for Lab07A

Program 7-23
line_graph5.py
part 2

Lab07A Graph

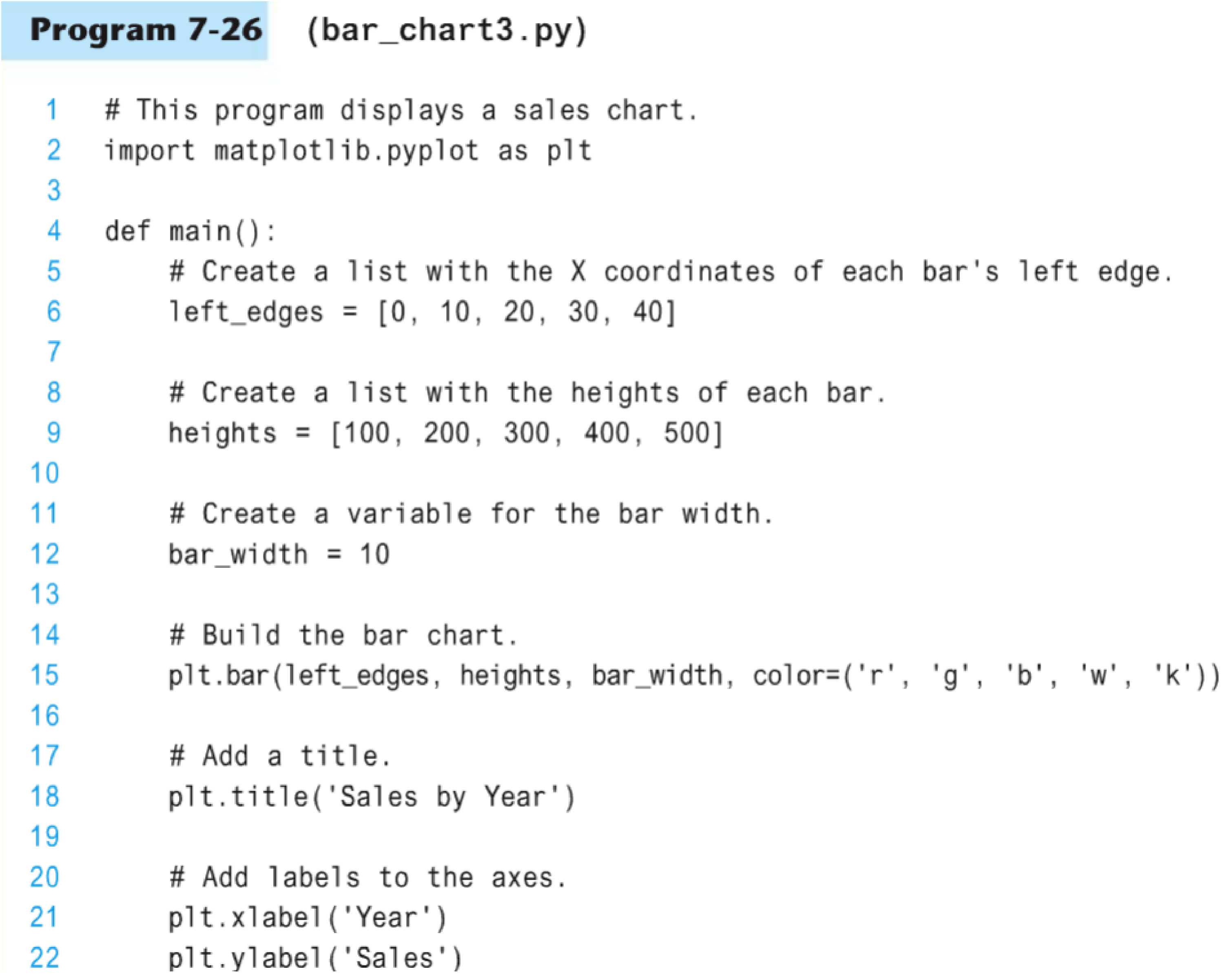


Lab07B

- Change the graph from a line graph to a Bar graph.
- Make the individual bar widths: BAR_WIDTH = 0.5
- The pseudocode for a Bar graph is essentially the same as for a line graph, the only difference being in the build command.
- Customize your xticks by using the names of the 12 months of the year: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
- When you have your bar chart displayed, call the instructor over so that you may be given proper credit.

Lab07B Pseudocode

- Prepare the **gas** list.
  - Open the text file 1994_Weekly_Gas_Average.txt .
  - Read the contents into a list called **gas** .
  - Close the file.
  - Strip the newline from each element.
- Prepare the **x_coords** list.
  - Create a list with 52 elements, with the numbers [1, 2, 3, …, 52]
  - Name the list **x_coords .**
- Plot the bar graph.
  - Build the bar graph
  - Limit the X axis to the range 1-52.
  - Add a Title (e.g. "1994 Weekly Gas Prices")
  - Add labels to the X axis
  - Add labels to the Y axis
  - Customize your xticks by using the names of the 12 months of the year:
    - ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
  - Display the graph.



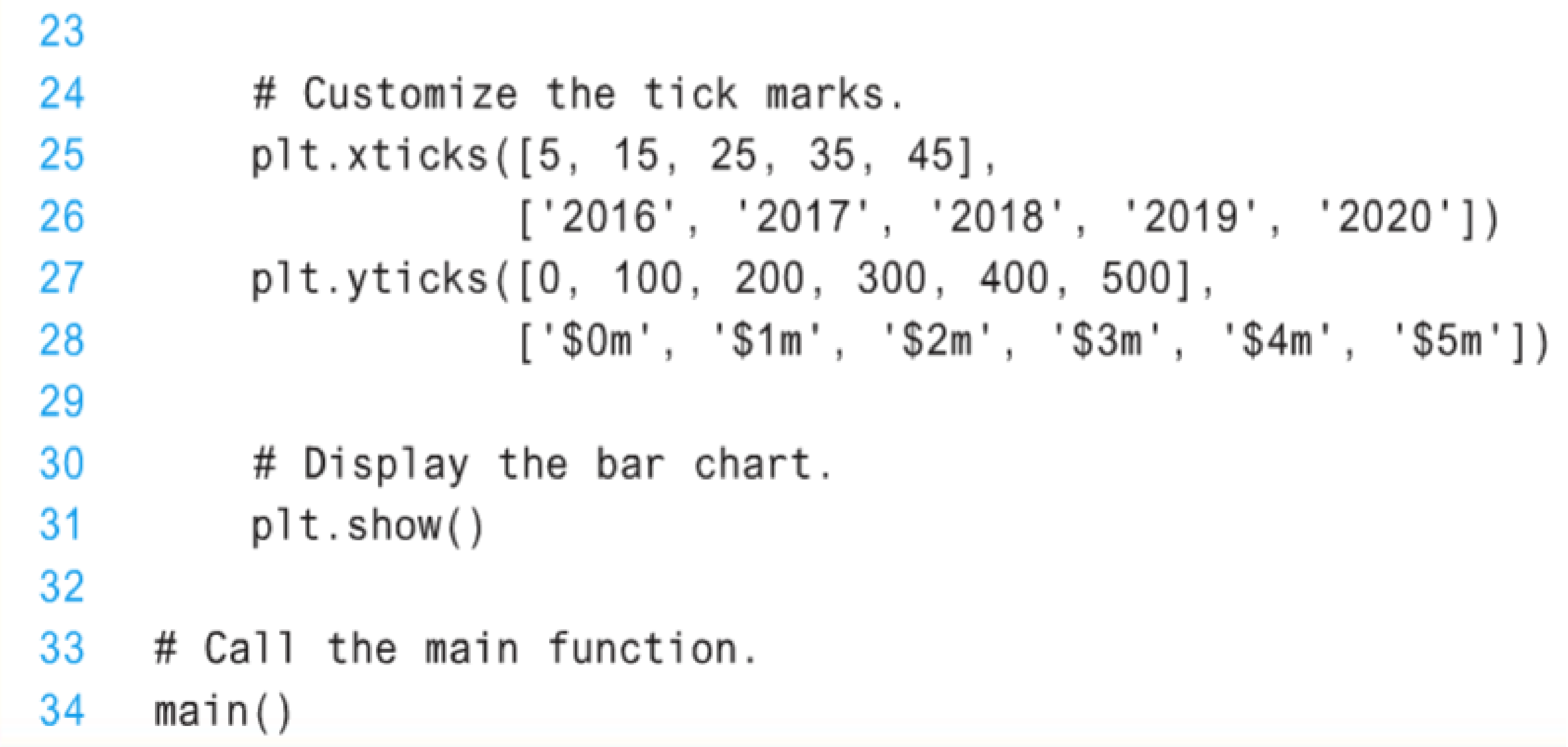
Program 7-26 (bar_chart3.py)

```python
# This program displays a sales chart.
import matplotlib.pyplot as plt

def main():
    # Create a list with the X coordinates of each bar's left edge.
    left_edges = [0, 10, 20, 30, 40]

    # Create a list with the heights of each bar.
    heights = [100, 200, 300, 400, 500]

    # Create a variable for the bar width.
    bar_width = 10

    # Build the bar chart.
    plt.bar(left_edges, heights, bar_width, color=('r', 'g', 'b', 'w', 'k'))

    # Add a title.
    plt.title('Sales by Year')

    # Add labels to the axes.
    plt.xlabel('Year')
    plt.ylabel('Sales')
```

Sample for Lab07B

Program 7-26
bar_chart3.py
part 1

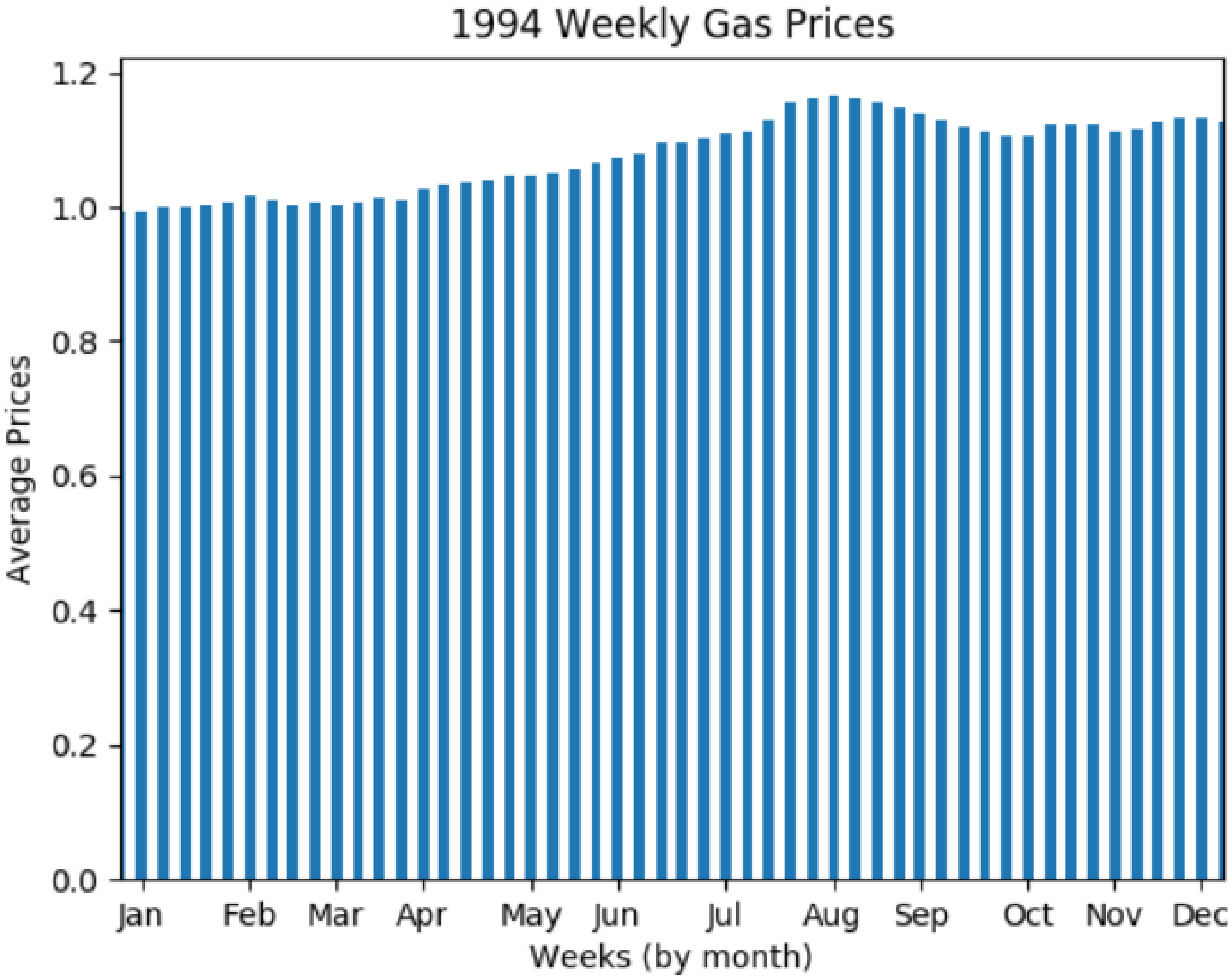
Sample for Lab07B



```
    # Customize the tick marks.
    plt.xticks([5, 15, 25, 35, 45],
               ['2016', '2017', '2018', '2019', '2020'])
    plt.yticks([0, 100, 200, 300, 400, 500],
               ['$0m', '$1m', '$2m', '$3m', '$4m', '$5m'])

    # Display the bar chart.
    plt.show()

# Call the main function.
main()
```

Program 7-26 bar_chart3.py part 2

Lab07B Graph

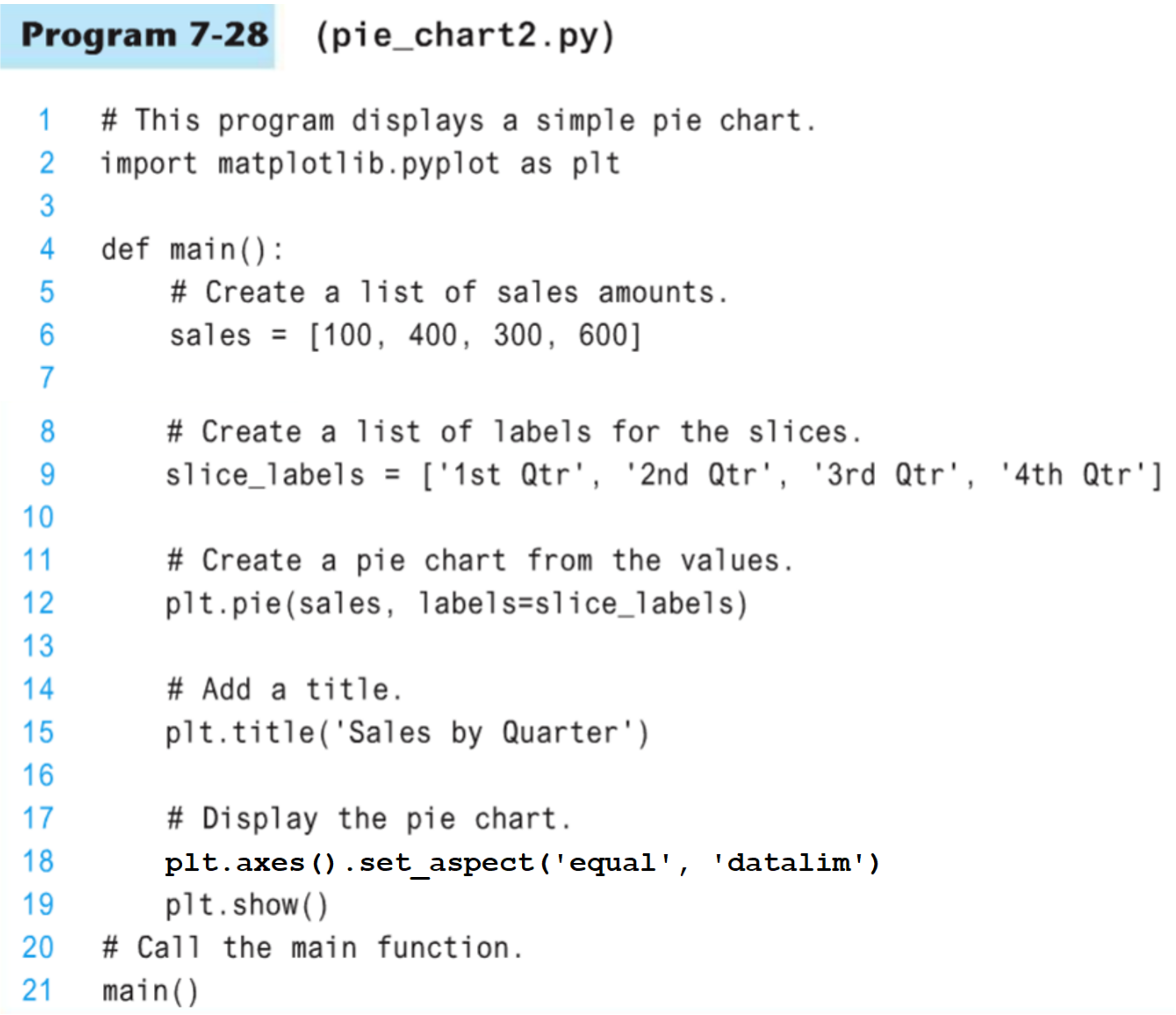


Lab07C

- The text file, **expenses.txt**, contains the amounts spent in the last month, in the following order:
  - Rent
  - Gas
  - Food
  - Clothing
  - Car Payment
  - Misc
- Write a Python program that reads the data from the file and uses matplotlib to plot a pie chart showing how the money was spent.
- Be sure to include the line that preserves an equal aspect ratio:
  - plt.axes().set_aspect('equal', 'datalim’)
- When you have finished, call the instructor over so that you may receive proper credit for your work.

Lab07C Pseudocode

- Prepare the **expenses** list.
    - Open the text file **expenses.txt** .
    - Read the contents into a list called **expenses** .
    - Close the file.
    - Strip the newline from each element.
- Prepare the **slice_labels** list.
    - Create a list of each of the expense categories in the order given.
        - ['Rent', 'Gas', 'Food', 'Clothing', 'Car Payment', 'Misc']
    - Name the list **slice_labels.**
- Plot the pie graph.
    - Create the pie chart from the lists you've created
    - Add a Title (e.g. "Monthly Expenses")
    - Include the line that preserves the equal aspect ratio
    - Display the graph.



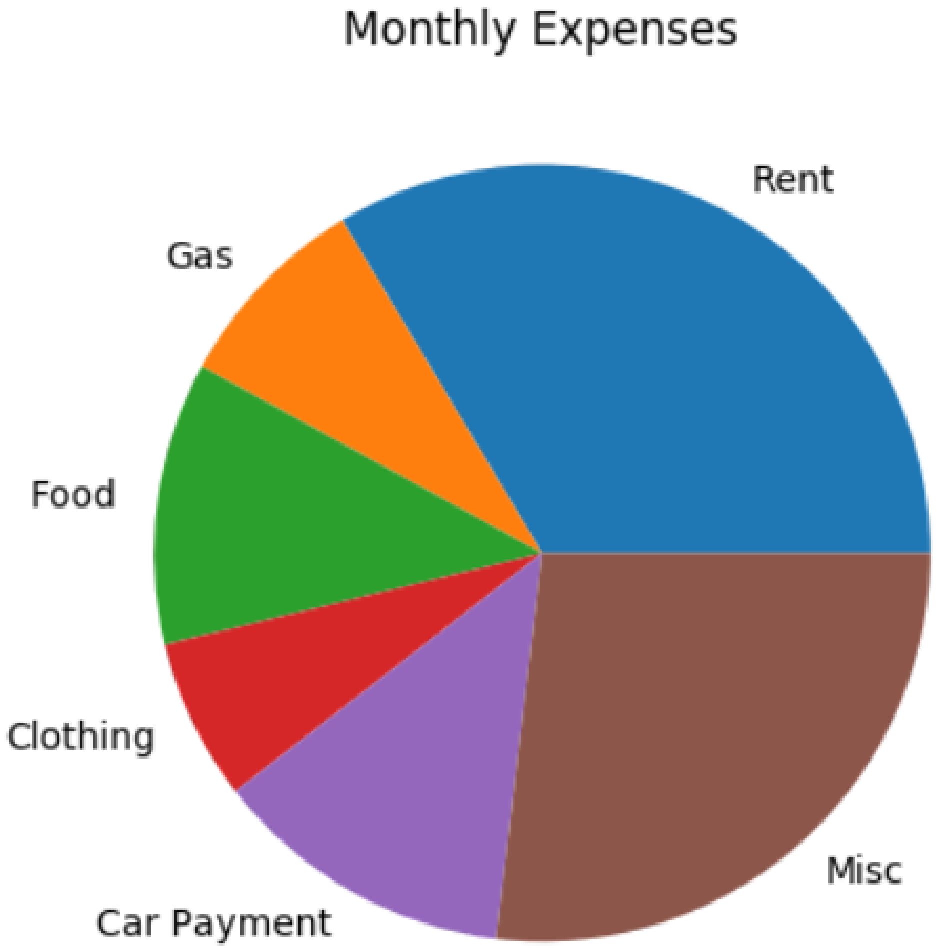
Program 7-28 (pie_chart2.py)

```python
# This program displays a simple pie chart.
import matplotlib.pyplot as plt

def main():
    # Create a list of sales amounts.
    sales = [100, 400, 300, 600]

    # Create a list of labels for the slices.
    slice_labels = ['1st Qtr', '2nd Qtr', '3rd Qtr', '4th Qtr']

    # Create a pie chart from the values.
    plt.pie(sales, labels=slice_labels)

    # Add a title.
    plt.title('Sales by Quarter')

    # Display the pie chart.
    plt.axes().set_aspect('equal', 'datalim')
    plt.show()
# Call the main function.
main()
```

Sample for Lab07C

Program 7-28
pie_chart2.py

Lab07C Graph


Monthly Expenses
Rent
Gas
Food
Clothing
Car Payment
Misc