3. Write code that does the following: opens an output file with the filename `number_list.txt`, uses a loop to write the numbers 1 through 100 to the file, then closes the file.
4. Write code that does the following: opens the `number_list.txt` file that was created by the code you wrote in question 3, reads all of the numbers from the file and displays them, then closes the file.
5. Modify the code that you wrote in problem 4 so it adds all of the numbers read from the file and displays their total.
6. Write code that opens an output file with the filename `number_list.txt`, but does not erase the file's contents if it already exists.
7. A file exists on the disk named `students.txt`. The file contains several records, and each record contains two fields: (1) the student's name, and (2) the student's score for the final exam. Write code that deletes the record containing "John Perz" as the student name.
8. A file exists on the disk named `students.txt`. The file contains several records, and each record contains two fields: (1) the student's name, and (2) the student's score for the final exam. Write code that changes Julie Milan's score to 100.
9. What will the following code display?

```
try:
  x = float('abc123')
  print('The conversion is complete.')
except IOError:
  print('This code caused an IOError.')
except ValueError:
  print('This code caused a ValueError.')
print('The end.')
```

10. What will the following code display?

```
try:
  x = float(abc123)
  print(x)
except ValueError:
  print('This code caused a ValueError.')
except TypeError:
  print('This code caused a TypeError.')
except NameError:
  print('This code caused a NameError.')
print('The end.')
```

Programming Exercises


VideoNote
File Display

1. File Display

Assume a file containing a series of integers is named `numbers.txt` and exists on the computer's disk. Write a program that displays all of the numbers in the file.

2. File Head Display

Write a program that asks the user for the name of a file. The program should display only the first five lines of the file's contents. If the file contains less than five lines, it should display the file's entire contents.

3. Line Numbers

Write a program that asks the user for the name of a file. The program should display the contents of the file with each line preceded with a line number followed by a colon. The line numbering should start at 1.

4. High Score

Assume that a file named `scores.txt` exists on the computer's disk. It contains a series of records, each with two fields – a name, followed by a score (an integer between 1 and 100). Write a program that displays the name and score of the record with the highest score, as well as the number of records in the file. *(Hint: Use a variable and an "if" statement to keep track of the highest score found as you read through the records, and a variable to keep count of the number of records.)*

5. Sum of Numbers

Assume a file containing a series of integers is named `numbers.txt` and exists on the computer's disk. Write a program that reads all of the numbers stored in the file and calculates their total.

6. Average of Numbers

Assume a file containing a series of integers is named `numbers.txt` and exists on the computer's disk. Write a program that calculates the average of all the numbers stored in the file.

7. Word List File Writer

Write a program that asks the user how many words they would like to write to a file, and then asks the user to enter that many words, one at a time. The words should be written to a file.

8. Word List File Reader

This exercise assumes you have completed the Programming Exercise 7, *Word List File Writer*. Write another program that reads the words from the file and displays the following data:

- The number of words in the file.
- The longest word in the file.
- The average length of all of the words in the file.

9. Exception Handing

Modify the program that you wrote for Exercise 6 so it handles the following exceptions:

- It should handle any `IOError` exceptions that are raised when the file is opened and data is read from it.
- It should handle any `ValueError` exceptions that are raised when the items that are read from the file are converted to a number.

10. Golf Scores

The Springfork Amateur Golf Club has a tournament every weekend. The club president has asked you to write two programs:

1. A program that will read each player's name and golf score as keyboard input, then save these as records in a file named `golf.txt`. (Each record will have a field for the player's name and a field for the player's score.)
2. A program that reads the records from the `golf.txt` file and displays them.

11. Personal Web Page Generator

Write a program that asks the user for his or her name, then asks the user to enter a sentence that describes himself or herself. Here is an example of the program's screen:

```
Enter your name: Julie Taylor Enter
Describe yourself: I am a computer science major, a member of the
Jazz club, and I hope to work as a mobile app developer after I
graduate. Enter
```

Once the user has entered the requested input, the program should create an HTML file, containing the input, for a simple Web page. Here is an example of the HTML content, using the sample input previously shown:

```
<html>
<head>
</head>
<body>
   <center>
      <h1>Julie Taylor</h1>
   </center>
   <hr />
   I am a computer science major, a member of the Jazz club,
   and I hope to work as a mobile app developer after I graduate.
   <hr />
</body>
</html>
```

12. Average Steps Taken

A Personal Fitness Tracker is a wearable device that tracks your physical activity, calories burned, heart rate, sleeping patterns, and so on. One common physical activity that most of these devices track is the number of steps you take each day.

If you have downloaded this book's source code from the Premium Companion Website, you will find a file named `steps.txt` in the Chapter 06 folder. (The Premium Companion Website can be found at www.pearsonglobaleditions.com/gaddis.) The `steps.txt` file contains the number of steps a person has taken each day for a year. There are 365 lines in the file, and each line contains the number of steps taken during a day. (The first line is the number of steps taken on January 1st, the second line is the number of steps taken on January 2nd, and so forth.) Write a program that reads the file, then displays the average number of steps taken for each month. (The data is from a year that was not a leap year, so February has 28 days.)