Homework 06

COMSC-122

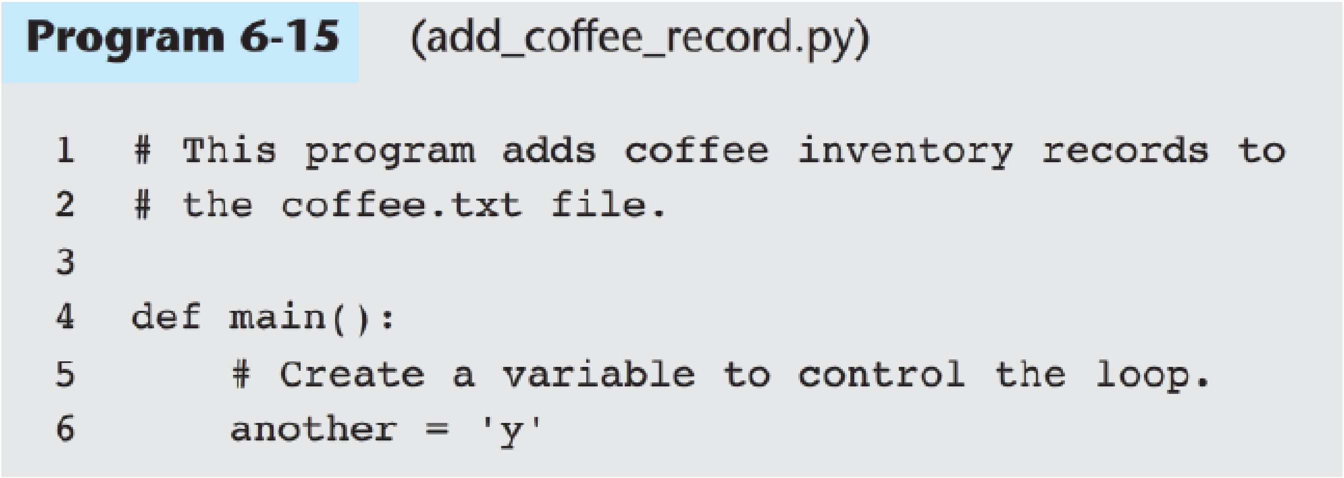
Fall 2017

Homework 6: Beginning of File Management System

- Program6-15 through Program6-19 show the makings of a complete File Management System.
- In Homework06, you will create a File Management System that will monitor and control coffee inventory.
- We will start off by demonstrating how to create the modules you will need by helping you do the first module.
- With Program6-15 we will do two things:
  - We will re-write Program6-15 as a Module called *LastnameFirstInitial*_coffee_records.py
    - For example, if your name was Juan Valdez, then for: *LastnameFirstInitial* you would put ValdezJ.
    - So your Module would have the name: ValdezJ_coffee_records.py
  - We will create a driver program which will run this module called: *YourName*-Hwrk6.py
    - You may want to refer to Program 5-28 as a sample of a menu driven program that uses Modules.

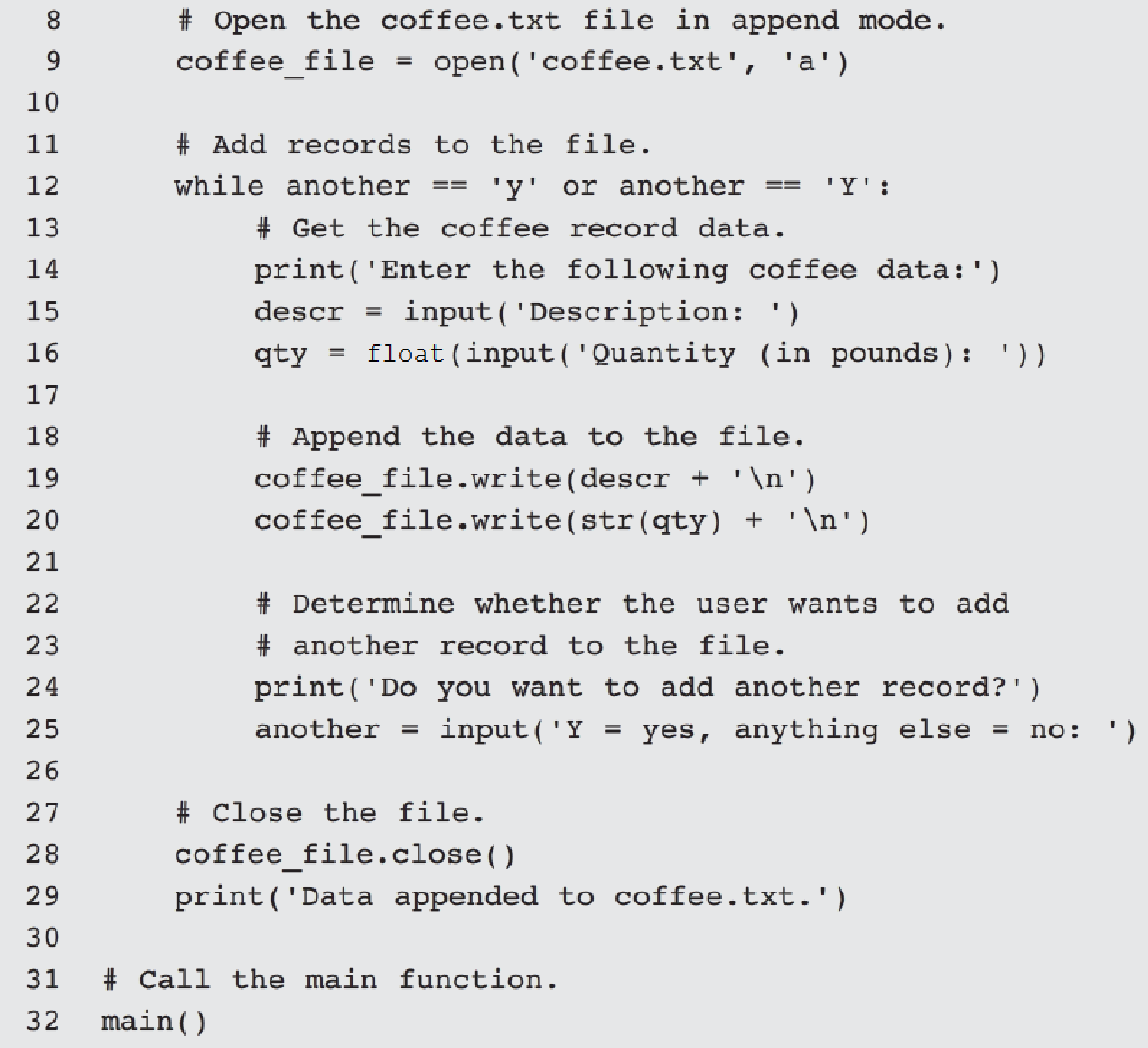
Program 6-15

- If we look closely at Program 6-15, we will notice that there are only two things that we must change if we wish to make it into a Module containing a function that we will use in the driver program.
    - First we must change the name of the **main()** function to another name. We will use the name: **add_coffee()** as the name of what was called **main().**
    - The second thing we must do is to delete the very last line of this program (line 32, main()), as this module will not call this function, rather it will be the driver program which calls this function.
- Once we have made these two changes, then the program which was labelled **Program6-15,** we will call module: ***LastnameFirstInitial*_coffee_records.py** , and it will contain a function, **add_coffee()**, that we will be using in the Driver program.



Program 6-15 (add_coffee_record.py)

```
1  # This program adds coffee inventory records to
2  # the coffee.txt file.
3
4  def main():
5      # Create a variable to control the loop.
6      another = 'y'
```

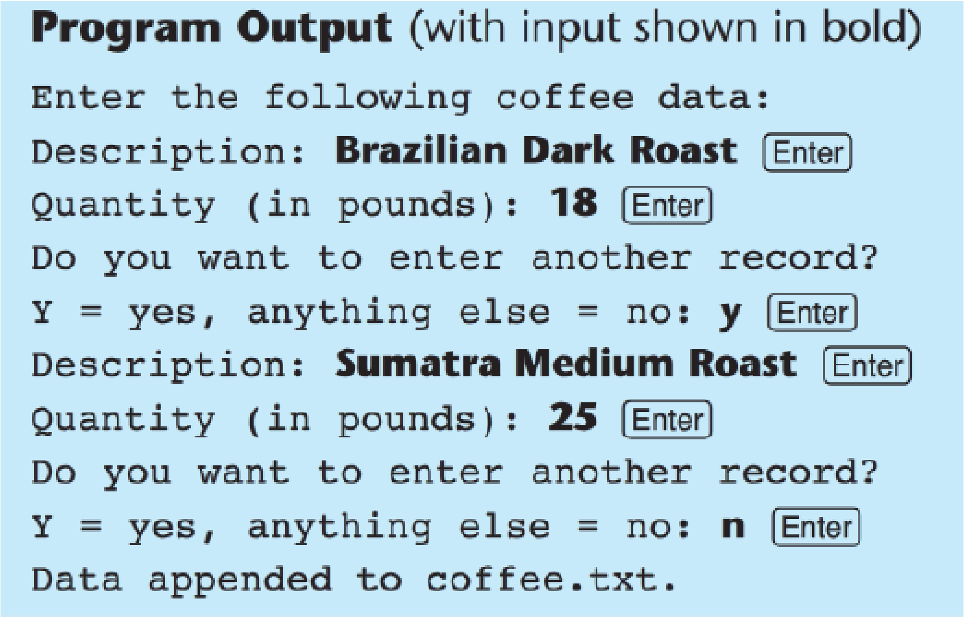


```
    # Open the coffee.txt file in append mode.
    coffee_file = open('coffee.txt', 'a')

    # Add records to the file.
    while another == 'y' or another == 'Y':
        # Get the coffee record data.
        print('Enter the following coffee data:')
        descr = input('Description: ')
        qty = float(input('Quantity (in pounds): '))

        # Append the data to the file.
        coffee_file.write(descr + '\n')
        coffee_file.write(str(qty) + '\n')

        # Determine whether the user wants to add
        # another record to the file.
        print('Do you want to add another record?')
        another = input('Y = yes, anything else = no: ')

    # Close the file.
    coffee_file.close()
    print('Data appended to coffee.txt.')

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

Program 6-15

Appending records to a File.

Program 6-15 Results



Program Output (with input shown in bold)

```
Enter the following coffee data:
Description: Brazilian Dark Roast Enter
Quantity (in pounds): 18 Enter
Do you want to enter another record?
Y = yes, anything else = no: y Enter
Description: Sumatra Medium Roast Enter
Quantity (in pounds): 25 Enter
Do you want to enter another record?
Y = yes, anything else = no: n Enter
Data appended to coffee.txt.
```

Homework 06 Cont.

- Now that we have one module, we are going to write the driver for this module: *YourName*-Hwrk06.py
- Initially, this driver will just run this first module. However, after you've got the first module working, you will be expanding this driver to include 4 additional modules.
- Below is the essence of the driving program which will run this File Management System to test and run the module *LastnameFirstInitial*_coffee_records.py

YourName-Lab06.py

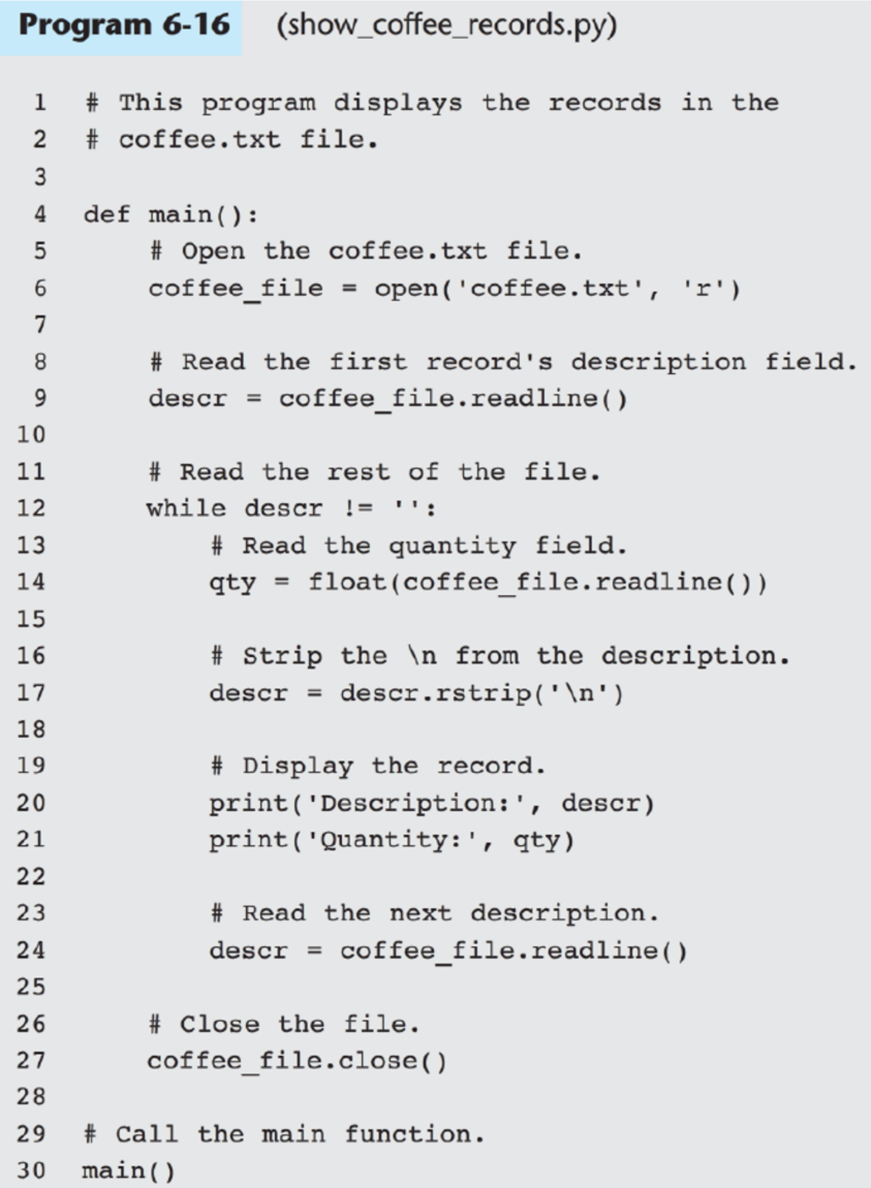
```
# Your Name
# A Coffee File Management Program

import ValdezJ_coffee_records       # Here’s where we import the module that we just made. But it should have your LastnameFirstInitial Not ValdezJ

ADD_COFFEE_CHOICE = 1               # Define Global constants for all your choices
QUIT_CHOICE = 6                     # Which includes the choice to quit

def main():
    choice = 0
    while choice != QUIT_CHOICE:
        display_menu()
        choice = int(input('Enter your choice: '))
        if choice == ADD_COFFEE_CHOICE:
            ValdezJ_coffee_records.add_coffee()       # We call the function add_coffee() contained in imported Module LastnameFirstInitial_coffee_records
        elif choice == QUIT_CHOICE:
            print('Exiting the program...')
        else:
            print('Error: invalid selection.')

def display_menu():
    print(' JUAN VALDEZ COFFEE MANAGEMENT MENU')
    print('1) Add more Coffee Choices to List')
    print('6) Quit')

main()
```

Homework 06 Cont.

- Once you have the first module up and running, now you will repeat what you did and add to the module, the second: Program 6-16, which contains the function **show_coffee()**.
- Then you will expand the driver program menu to include this additional module
    - You will also import this module
    - You will change the menu to include this module
- Once the second function is working, repeat the preceding 3 more times to include: Program 6-17, Program 6-18 and Program 6-19.
    - From these three programs you will be adding the following functions to your module: *LastnameFirstInitial*_coffee_records
        - **search_coffee()**
        - **modify_coffee()**
        - **delete_coffee()**
- Modify your driver program to accommodate these additions.

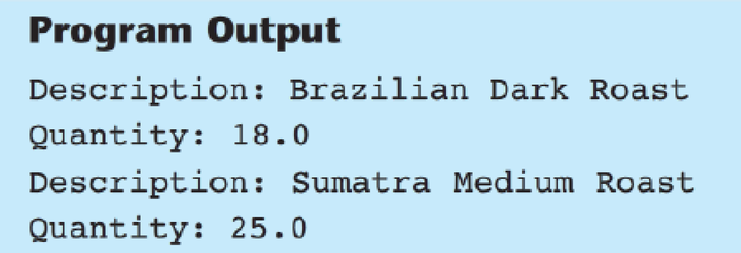
Program 6-16

Displaying all the records in a File.



Program 6-16 (show_coffee_records.py)

```python
# This program displays the records in the
# coffee.txt file.

def main():
    # Open the coffee.txt file.
    coffee_file = open('coffee.txt', 'r')

    # Read the first record's description field.
    descr = coffee_file.readline()

    # Read the rest of the file.
    while descr != '':
        # Read the quantity field.
        qty = float(coffee_file.readline())

        # Strip the \n from the description.
        descr = descr.rstrip('\n')

        # Display the record.
        print('Description:', descr)
        print('Quantity:', qty)

        # Read the next description.
        descr = coffee_file.readline()

    # Close the file.
    coffee_file.close()

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

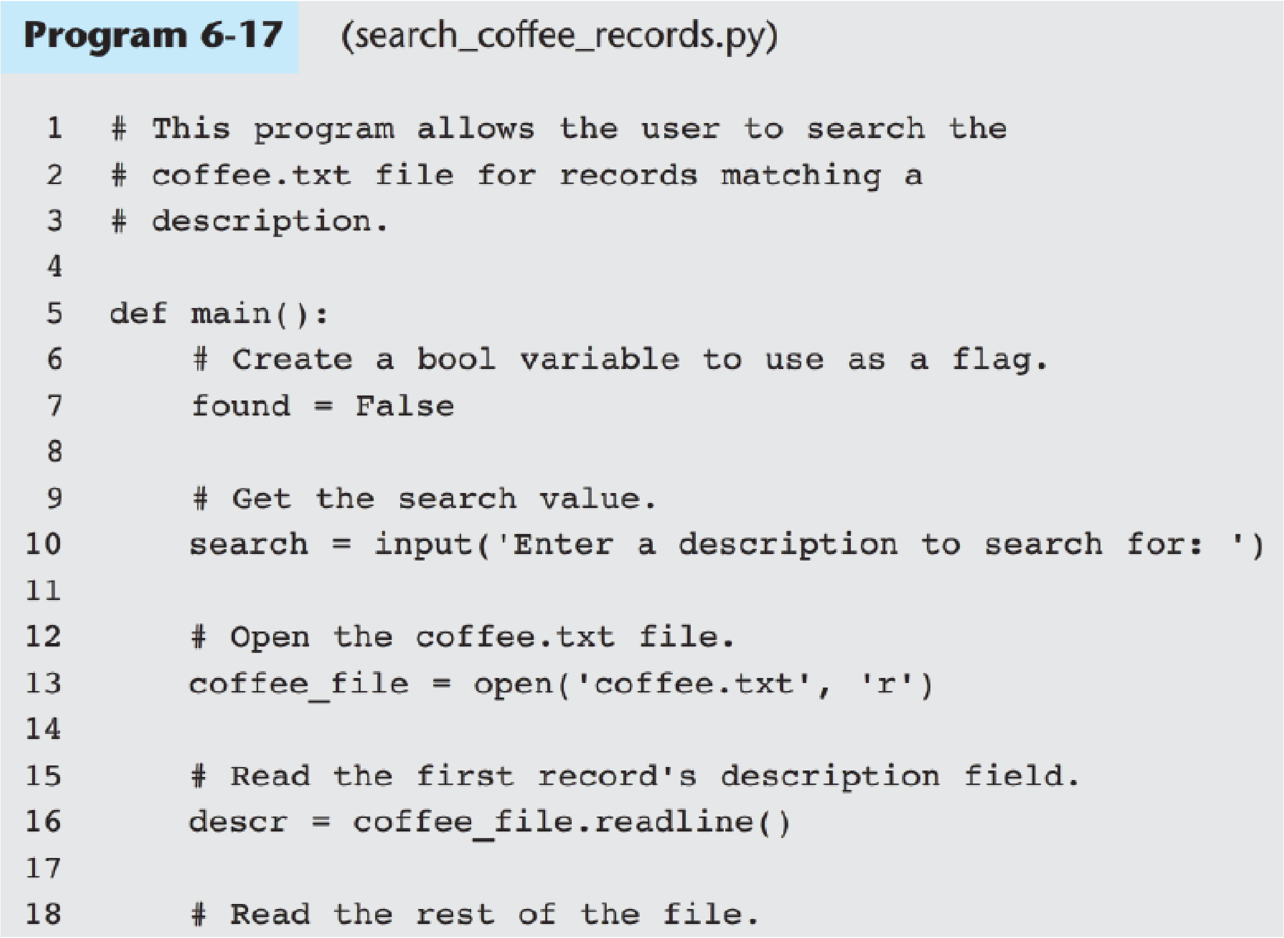


Program Output

```
Description: Brazilian Dark Roast
Quantity: 18.0
Description: Sumatra Medium Roast
Quantity: 25.0
```

Program 6-17

This program allows the user to make search of a file.

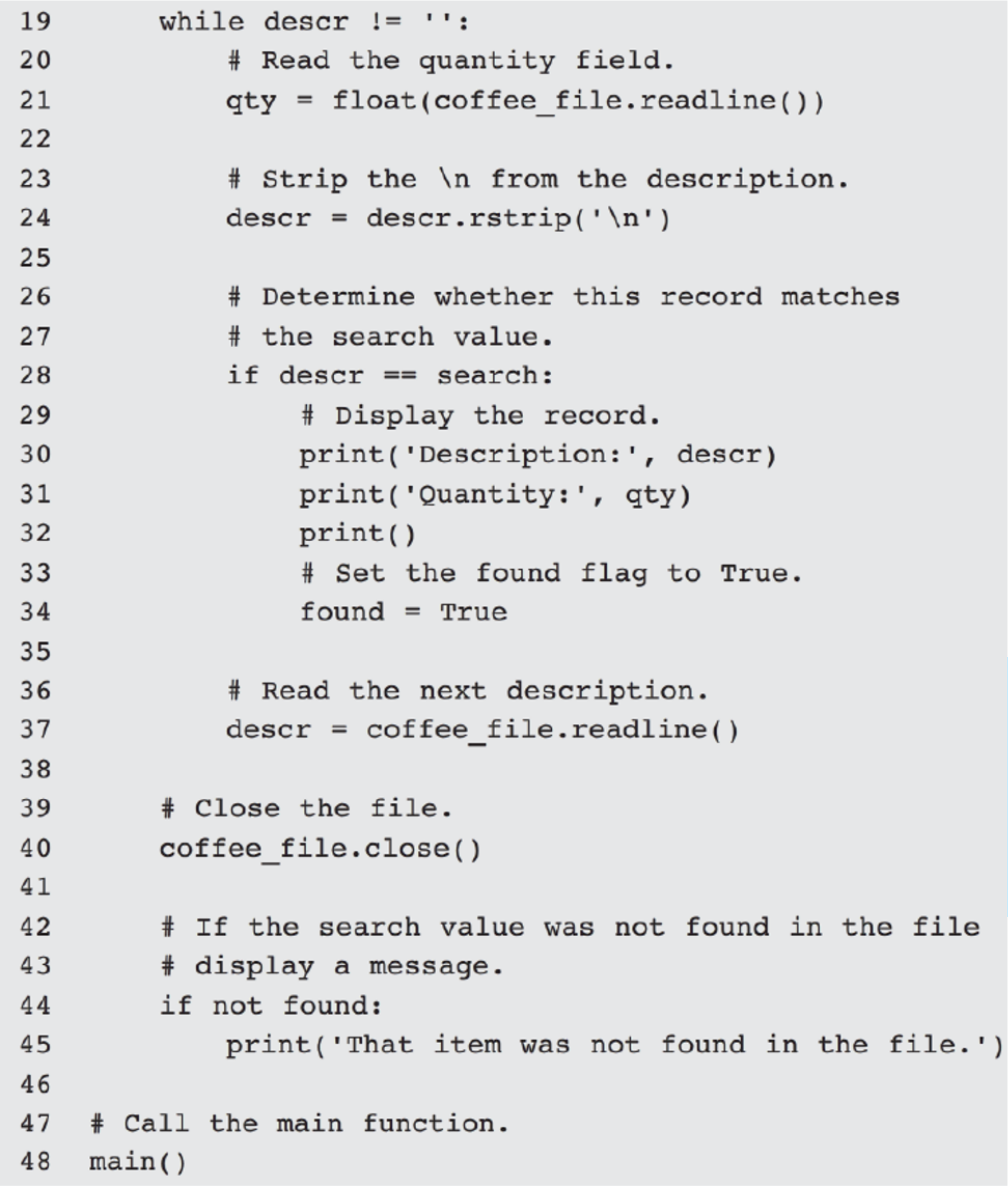


Program 6-17 (search_coffee_records.py)

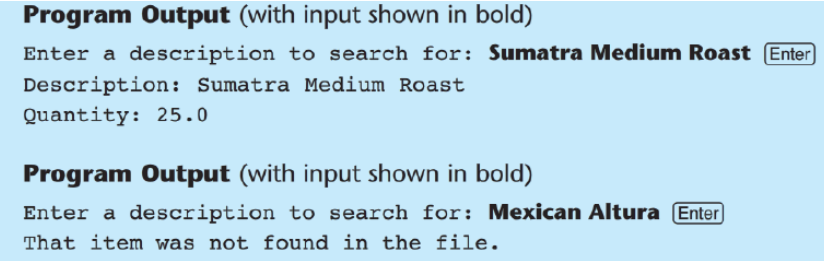
```
# This program allows the user to search the
# coffee.txt file for records matching a
# description.

def main():
    # Create a bool variable to use as a flag.
    found = False

    # Get the search value.
    search = input('Enter a description to search for: ')

    # Open the coffee.txt file.
    coffee_file = open('coffee.txt', 'r')

    # Read the first record's description field.
    descr = coffee_file.readline()

    # Read the rest of the file.
```

Program 6-17

Search cont.



```python
    while descr != '':
        # Read the quantity field.
        qty = float(coffee_file.readline())

        # Strip the \n from the description.
        descr = descr.rstrip('\n')

        # Determine whether this record matches
        # the search value.
        if descr == search:
            # Display the record.
            print('Description:', descr)
            print('Quantity:', qty)
            print()
            # Set the found flag to True.
            found = True

        # Read the next description.
        descr = coffee_file.readline()

    # Close the file.
    coffee_file.close()

    # If the search value was not found in the file
    # display a message.
    if not found:
        print('That item was not found in the file.')

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

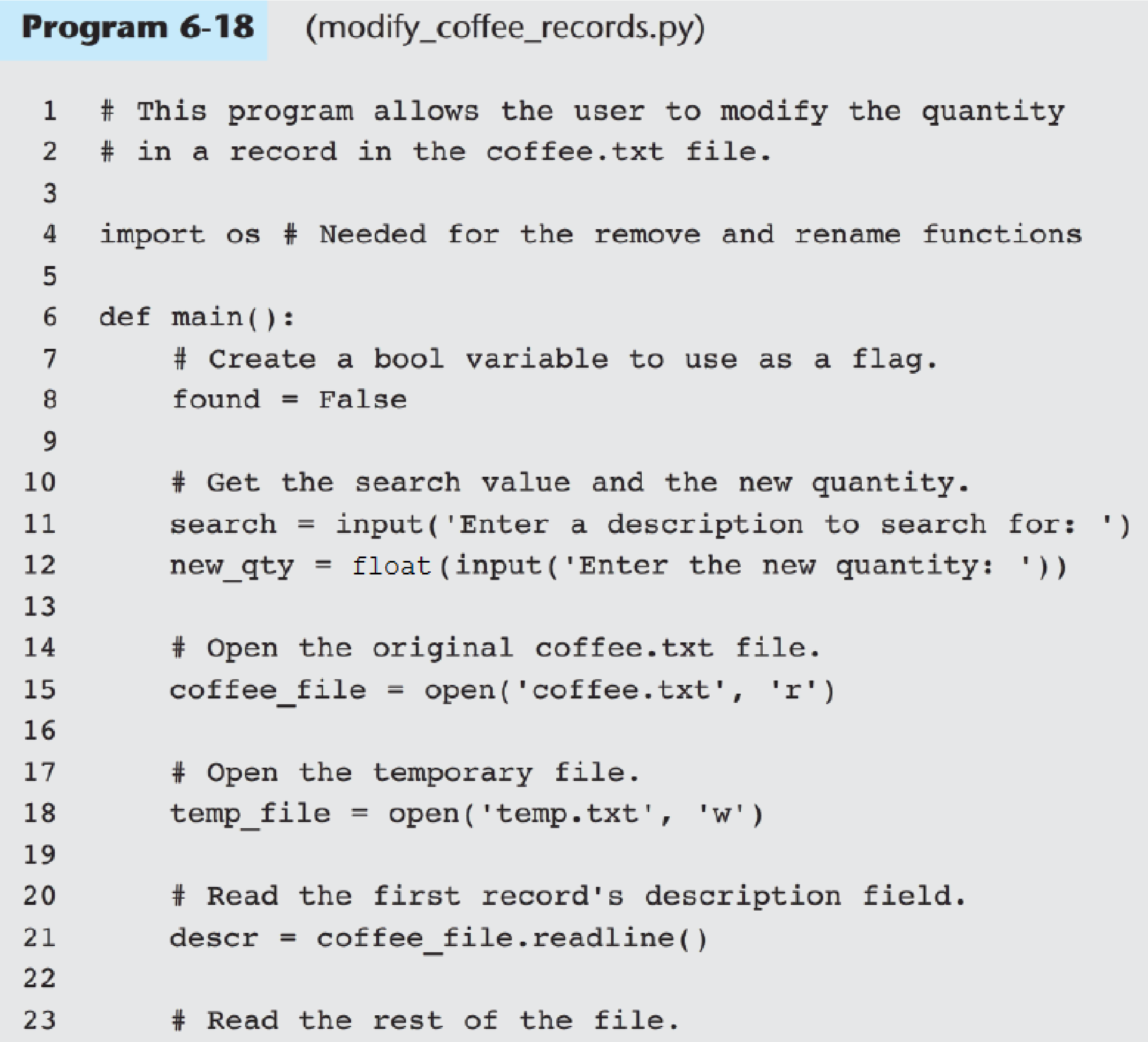


Program Output (with input shown in bold)

```
Enter a description to search for: Sumatra Medium Roast [Enter]
Description: Sumatra Medium Roast
Quantity: 25.0
```

Program Output (with input shown in bold)

```
Enter a description to search for: Mexican Altura [Enter]
That item was not found in the file.
```

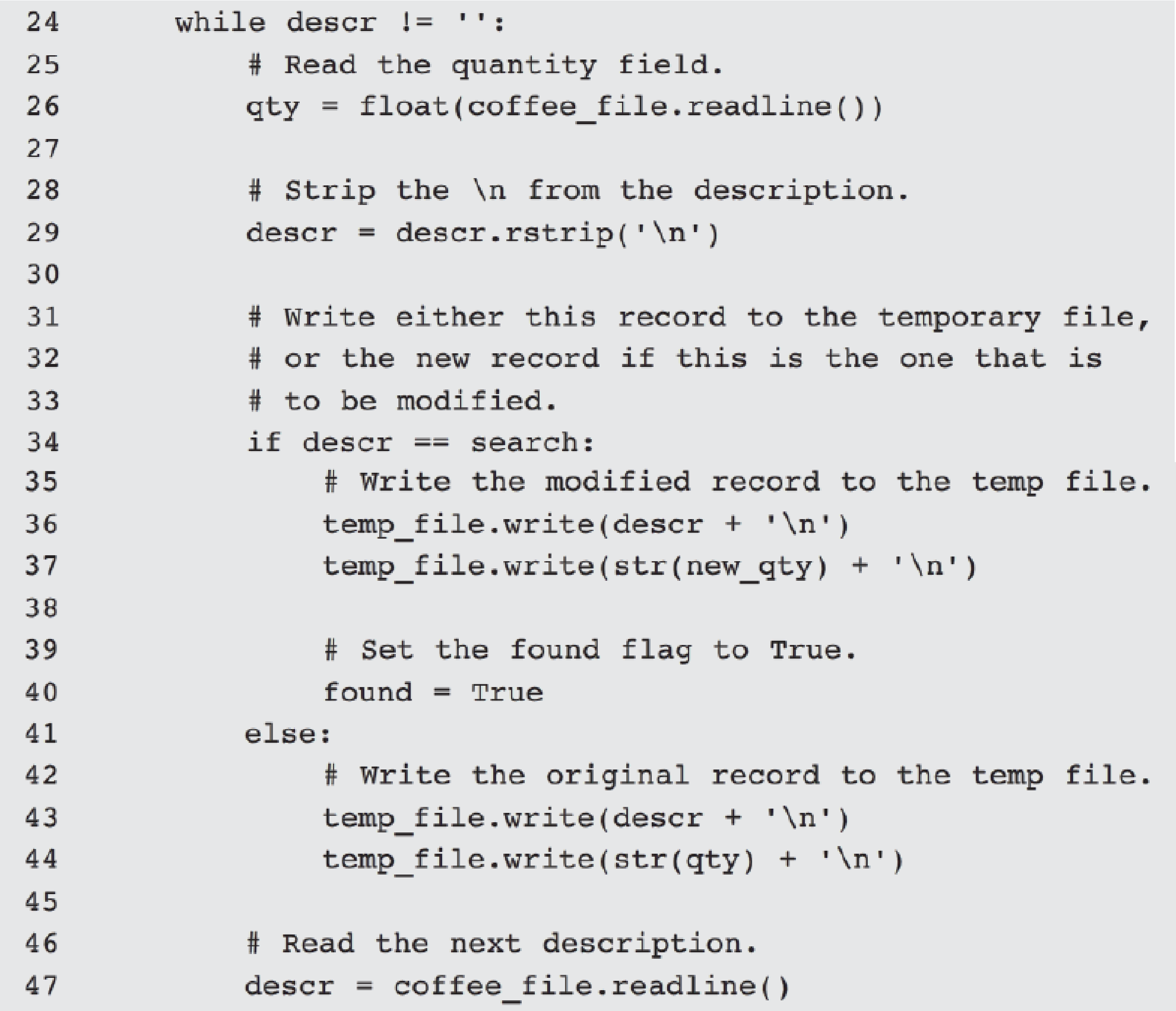


Program 6-18 (modify_coffee_records.py)

```python
# This program allows the user to modify the quantity
# in a record in the coffee.txt file.

import os # Needed for the remove and rename functions

def main():
    # Create a bool variable to use as a flag.
    found = False

    # Get the search value and the new quantity.
    search = input('Enter a description to search for: ')
    new_qty = float(input('Enter the new quantity: '))

    # Open the original coffee.txt file.
    coffee_file = open('coffee.txt', 'r')

    # Open the temporary file.
    temp_file = open('temp.txt', 'w')

    # Read the first record's description field.
    descr = coffee_file.readline()

    # Read the rest of the file.
```

Program 6-18

Editing a File



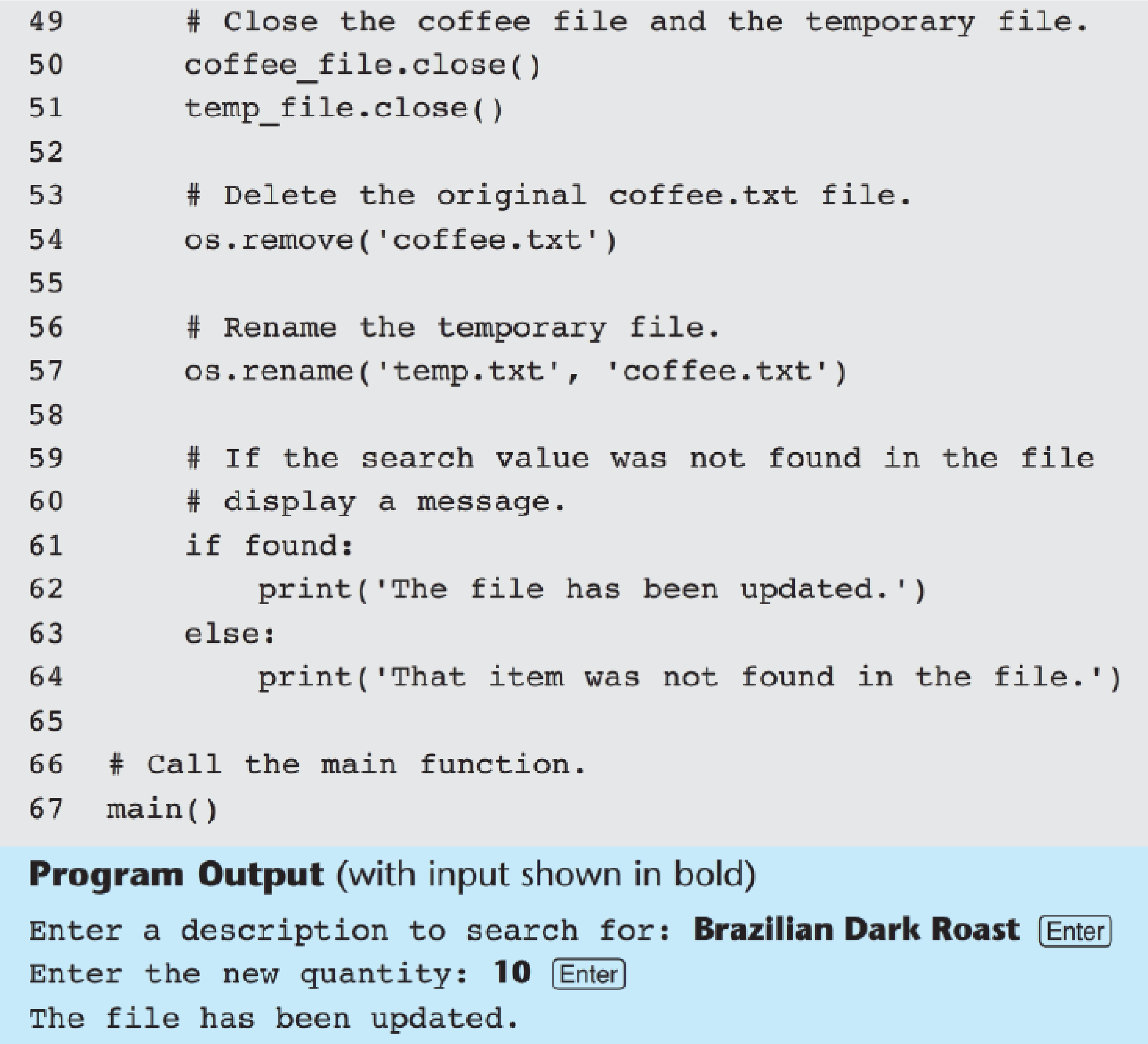
```
    while descr != '':
        # Read the quantity field.
        qty = float(coffee_file.readline())

        # Strip the \n from the description.
        descr = descr.rstrip('\n')

        # Write either this record to the temporary file,
        # or the new record if this is the one that is
        # to be modified.
        if descr == search:
            # Write the modified record to the temp file.
            temp_file.write(descr + '\n')
            temp_file.write(str(new_qty) + '\n')

            # Set the found flag to True.
            found = True
        else:
            # Write the original record to the temp file.
            temp_file.write(descr + '\n')
            temp_file.write(str(qty) + '\n')

        # Read the next description.
        descr = coffee_file.readline()
```

Program 6-18

Editing
a File cont.

Program 6-18

Editing a File concluded

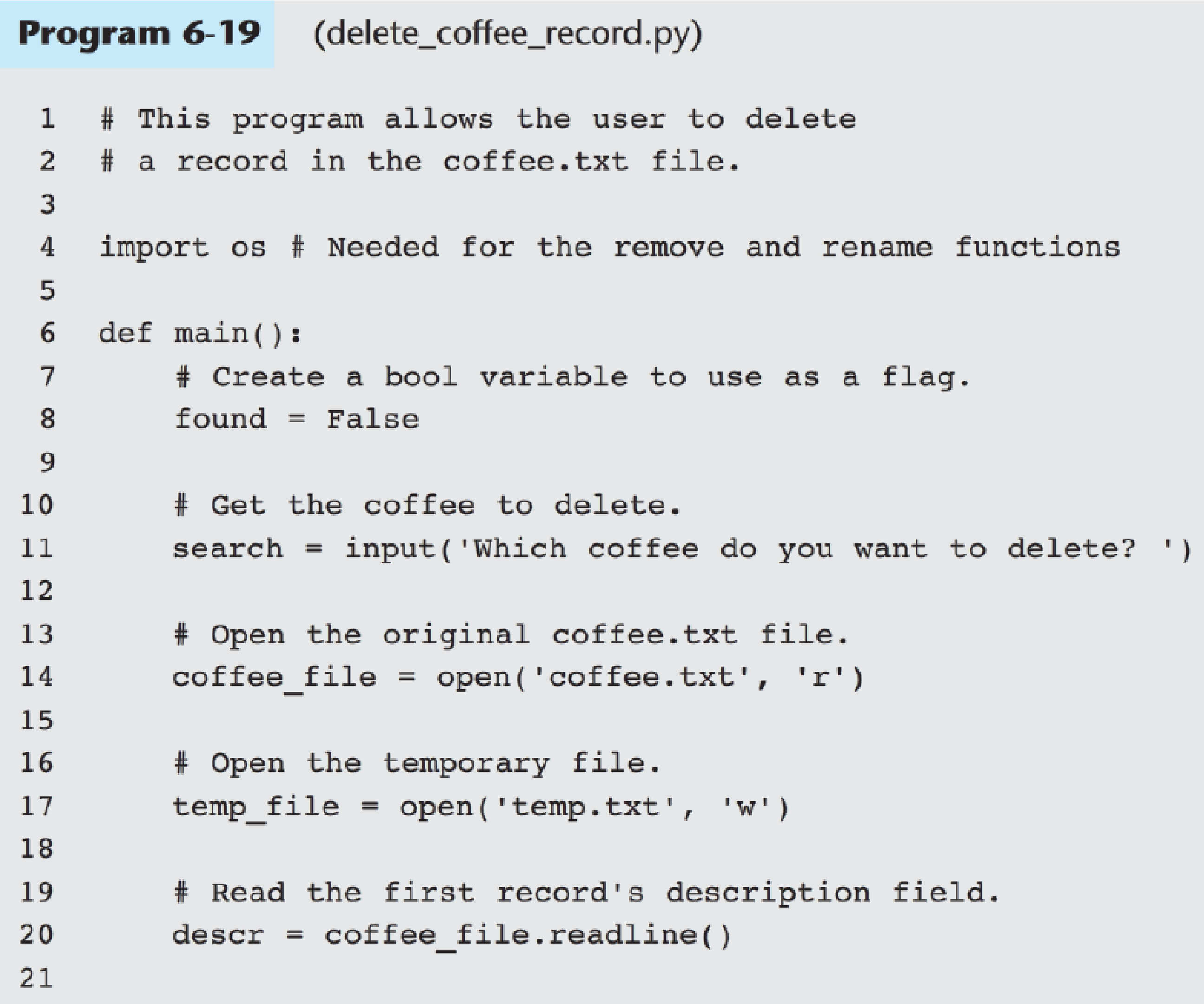


```python
    # Close the coffee file and the temporary file.
    coffee_file.close()
    temp_file.close()

    # Delete the original coffee.txt file.
    os.remove('coffee.txt')

    # Rename the temporary file.
    os.rename('temp.txt', 'coffee.txt')

    # If the search value was not found in the file
    # display a message.
    if found:
        print('The file has been updated.')
    else:
        print('That item was not found in the file.')

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

Program Output (with input shown in bold)

```
Enter a description to search for: Brazilian Dark Roast Enter
Enter the new quantity: 10 Enter
The file has been updated.
```

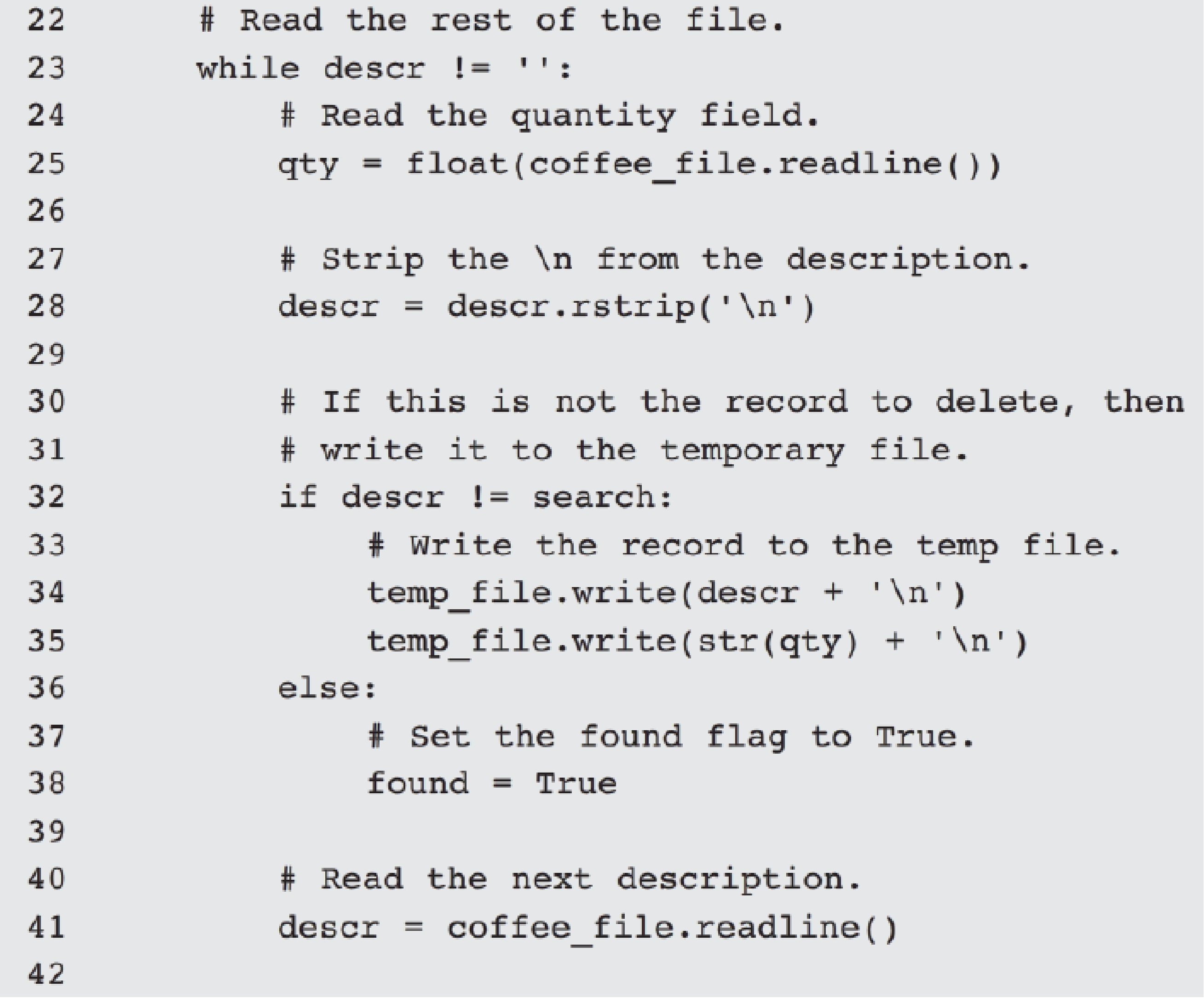
Program 6-19: Deleting a Record from a File



Program 6-19 (delete_coffee_record.py)

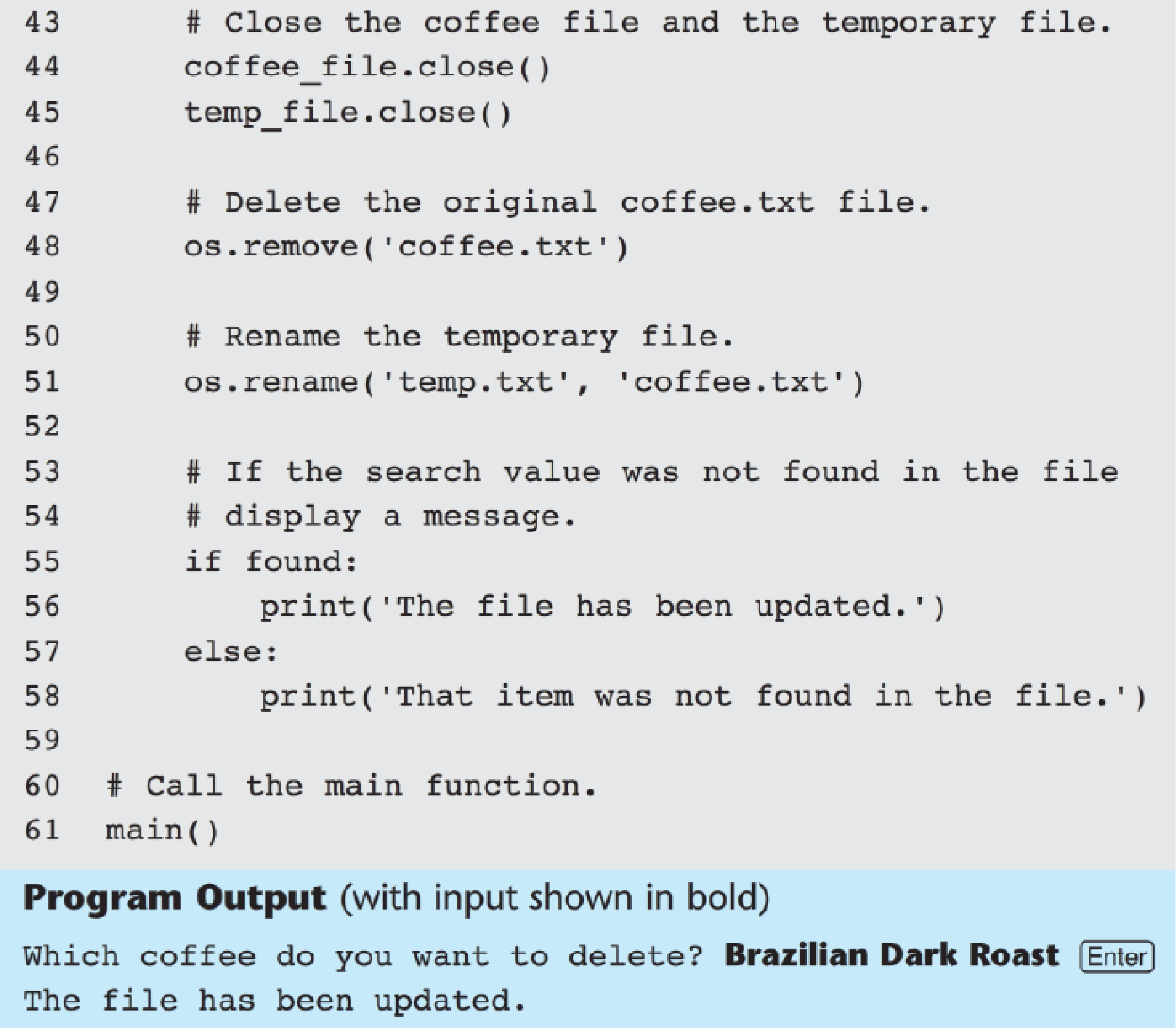
```python
# This program allows the user to delete
# a record in the coffee.txt file.

import os # Needed for the remove and rename functions

def main():
    # Create a bool variable to use as a flag.
    found = False

    # Get the coffee to delete.
    search = input('Which coffee do you want to delete? ')

    # Open the original coffee.txt file.
    coffee_file = open('coffee.txt', 'r')

    # Open the temporary file.
    temp_file = open('temp.txt', 'w')

    # Read the first record's description field.
    descr = coffee_file.readline()

```

Program 6-19: Deleting a Record from a File Cont.



```python
    # Read the rest of the file.
    while descr != '':
        # Read the quantity field.
        qty = float(coffee_file.readline())

        # Strip the \n from the description.
        descr = descr.rstrip('\n')

        # If this is not the record to delete, then
        # write it to the temporary file.
        if descr != search:
            # Write the record to the temp file.
            temp_file.write(descr + '\n')
            temp_file.write(str(qty) + '\n')
        else:
            # Set the found flag to True.
            found = True

        # Read the next description.
        descr = coffee_file.readline()

```

Program 6-19: Deleting a record from a file concluded



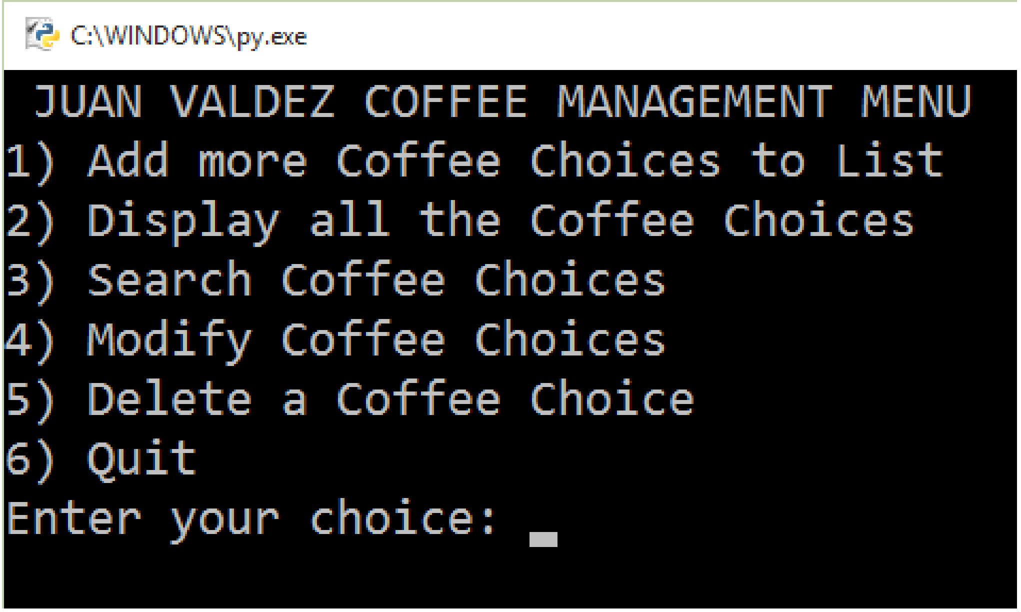
```
    # Close the coffee file and the temporary file.
    coffee_file.close()
    temp_file.close()

    # Delete the original coffee.txt file.
    os.remove('coffee.txt')

    # Rename the temporary file.
    os.rename('temp.txt', 'coffee.txt')

    # If the search value was not found in the file
    # display a message.
    if found:
        print('The file has been updated.')
    else:
        print('That item was not found in the file.')

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

Program Output (with input shown in bold)

```
Which coffee do you want to delete? Brazilian Dark Roast Enter
The file has been updated.
```

Homework 06 Concluded

- Here is what the Homework6 menu should look like.
  - However, substitute your name for that of Juan Valdez.



```
 JUAN VALDEZ COFFEE MANAGEMENT MENU
1) Add more Coffee Choices to List
2) Display all the Coffee Choices
3) Search Coffee Choices
4) Modify Coffee Choices
5) Delete a Coffee Choice
6) Quit
Enter your choice: _
```