CIS1400 – Programming Logic and Technique

Topic 8 → Data Files

Chapter Topics

10.1 Introduction to File Input and Output

10.2 Using Loops to Process Files

10.3 Using Files and Arrays

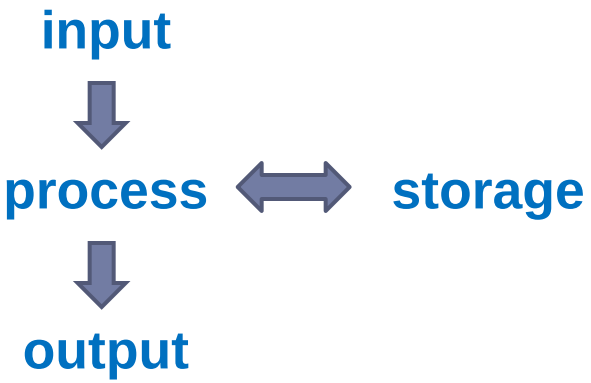
10.4 Processing Records

10.5 Control Break Logic

10.1 Introduction to File Input and Output

When a program needs to save data for later use, it writes the data in a file and can be used later

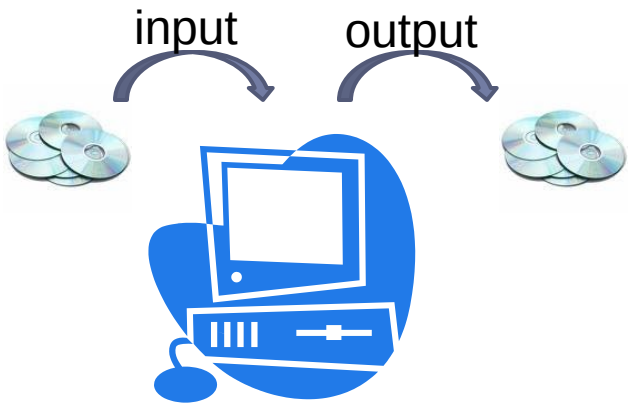
- In previous programs, internal data was stored in variables
- File input and output can interact with various types of applications
  - Word processors
  - Image editors
  - Spreadsheets
  - Games
  - Web browsers



10.1 Introduction to File Input and Output

Three steps must take place for file interaction with an application

1. Open the file:
   - Output → means creating and preparing it for writing data
   - Input → means opening a file and preparing it for reading data
2. Process the file:
   - Output → Writes data to file
   - Input → Reads data from file
3. Close the file:
   - Flush buffer contents (for write); disconnects from application

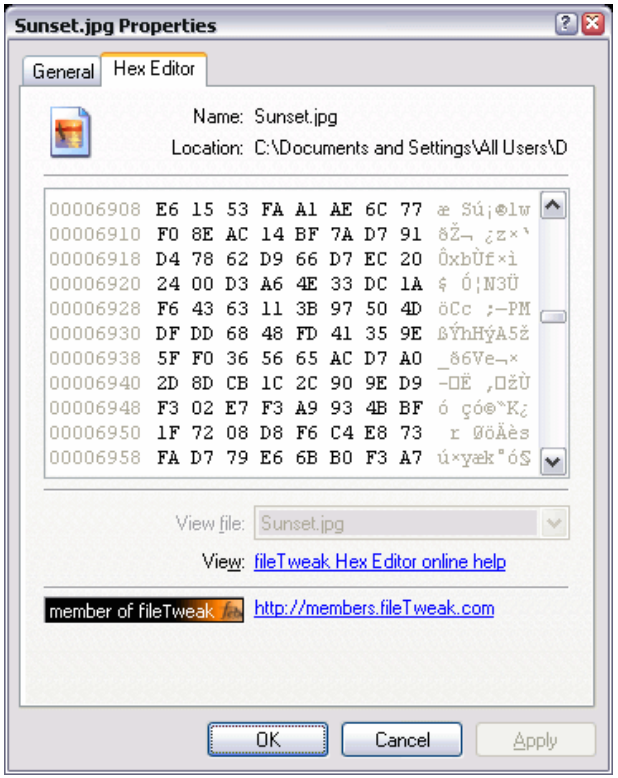


10.1 Introduction to File Input and Output

Types of files include text and binary

- A text file contains data that has been encoded as text, using ASCII or Unicode
  - Numbers in this type of file are stored as text
  - Use text editor (i.e. Notepad) to view
- A binary file contains data that has not been converted to text
  - Use hex or image editor to view

http://en.webhex.net/

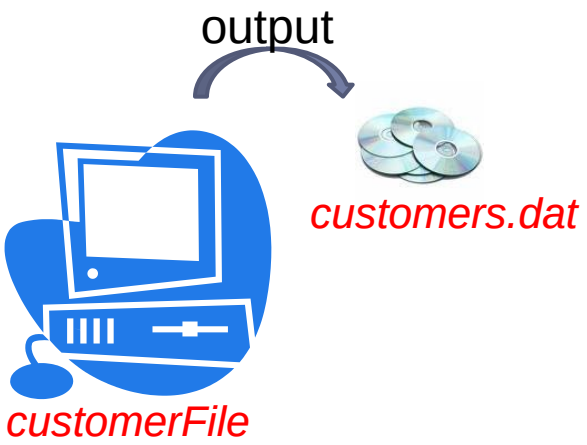


10.1 Introduction to File Input and Output

File data can be accessed in two ways:

- Sequential access
  - Data is accessed from the beginning to the end
  - All data must be read
- Direct access (aka random access)
  - Any piece of data can be accessed without reading the data that comes before or after it
  - Need byte location to access data item
- This chapter will focus on sequential access files

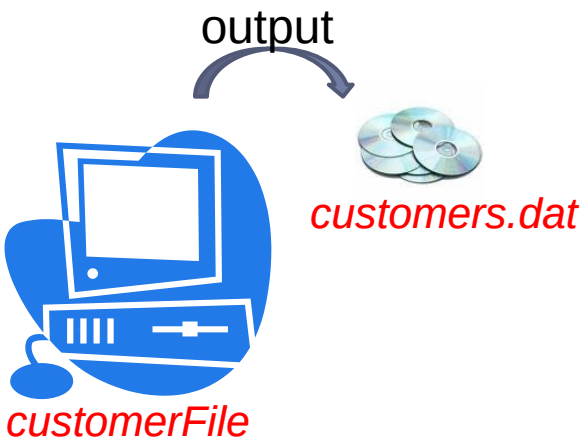
10.1 Introduction to File Input and Output



Create File and Write Data

1. Files that are created should be given an external name with an appropriate file extension
   - `customers.dat` where *`.dat`* represents general data
2. Must also create an internal name that is similar to a variable name
   - `Declare OutputFile customerFile`
     - *`OutputFile`* indicates the mode in which the file will be used
     - *`customerFile`* is the internal name used to work with the file

10.1 Introduction to File Input and Output



Create File and Write Data

3. Files must be opened

   link internal and external names

   ```
   Open customerFile "customers.dat"
   ```

4. Data can then be written to a file

   ```
   Write customerFile "Charles Pace"
   ```

   or

   ```
   Declare String name = "Charles Pace"
   Write customerFile name
   ```

5. Closing a file

   ```
   Close customerFile
   ```

10.1 Introduction to File Input and Output

Example: Program 10-1

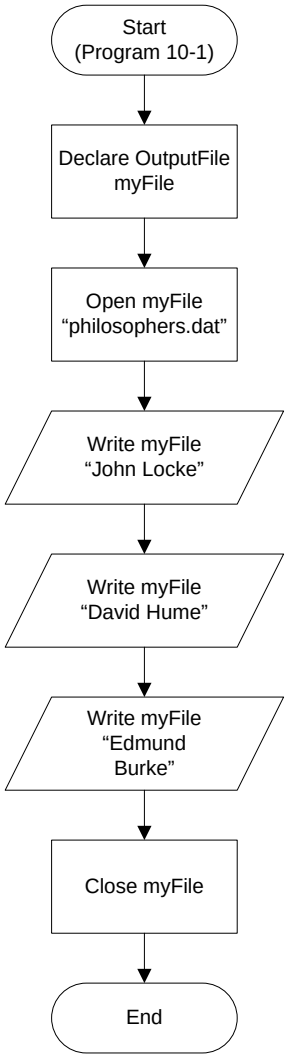
```
Declare OutputFile myFile
Open myFile "philosophers.dat"
Write myFile "John Locke"
Write myFile "David Hume"
Write myFile "Edmund Burke"
Close myFile
```

create file variable

open file

write to file

close file



VB example

10.1 Introduction to File Input and Output

Read Data From File

1. An internal variable must first be declared

```
Declare InputFile inventoryFile
```

   - *`InputFile`* indicates the mode in which the file will be used
   - *`inventoryFile`* is the internal name used to work with the file

2. The file can then be opened

   link internal and external names

```
Open inventoryFile "inventory.dat"
```

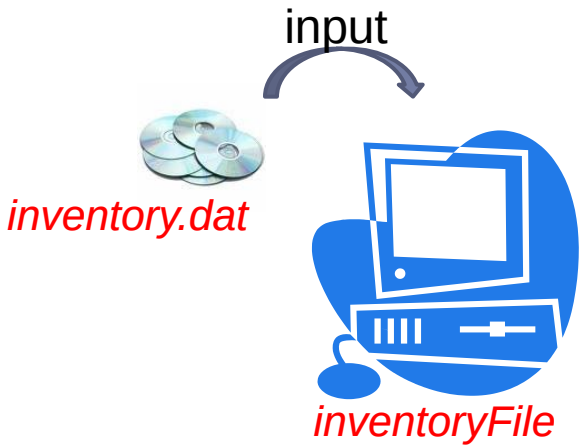
3. Data can then be read

   each read statement will update read position

```
Read inventoryFile itemName
```

4. Closing a file

```
Close inventoryFile
```



10.1 Introduction to File Input and Output

Example: Program 10-2

```
Declare InputFile myFile
Declare String name1,
     name2, name3
Open myFile "philosophers.dat"
Read myFile name1
Read myFile name2
Read myFile name3
Close myFile
Display "Here are the names ",
      "of three philosophers: "
Display name1
Display name2
Display name3
```

create file variable

open file

read from file

close file

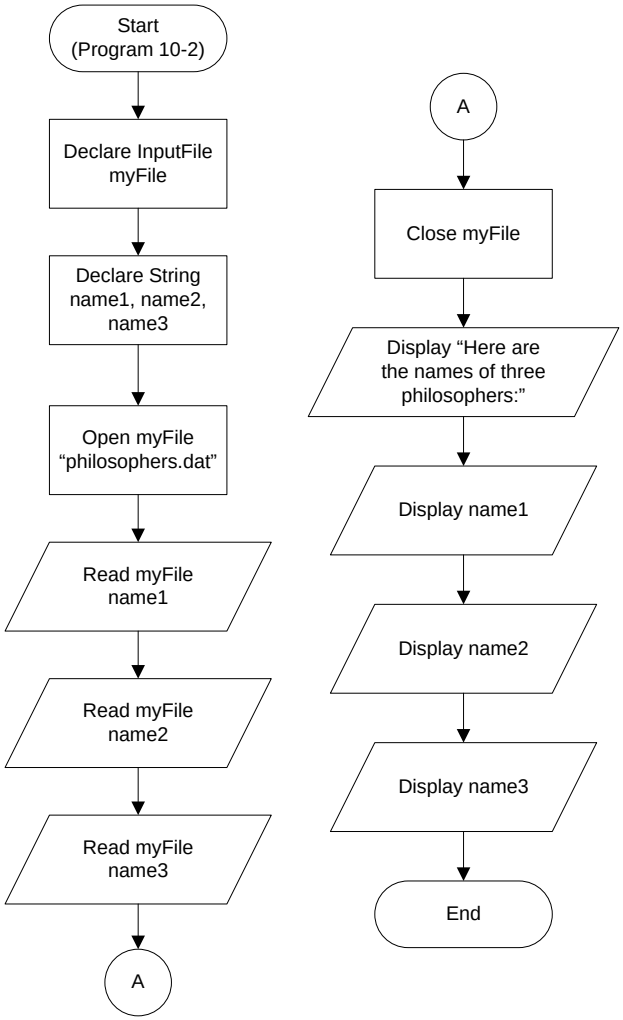
10.1 Introduction to File Input and Output

Example: Program 10-2

create file variable

open file

read from file



close file

VB example

10.1 Introduction to File Input and Output

The append mode

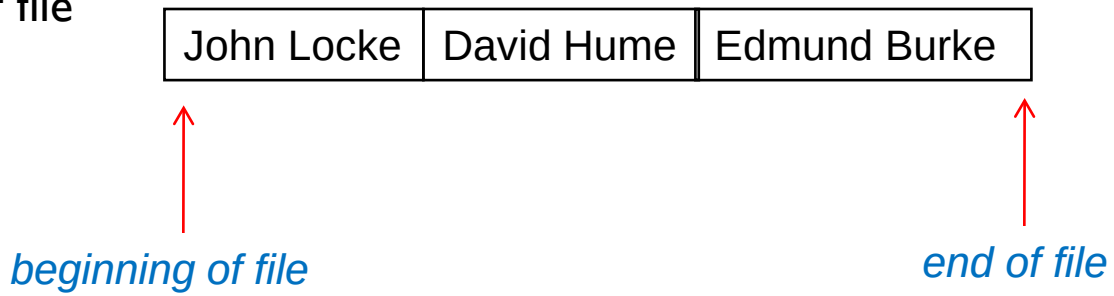
- In addition to read and write mode, append mode also exists to add data to a file that also exists
- If the file
  - does exist, it will not be erased, it will be added to
  - does not exist, it will be created
- When data is written to the file, it will be written to the end of the file
  - `Declare OutputFile AppendMode myFile`

VB example

10.1 Introduction to File Input and Output

Delimiters and EOF Marker

- A delimiter is a predefined character, or set of characters, that marks the end of each piece of data
  - It separates the different items stored in a file
- An End of File (EOF) marker is a special character or set of characters written to the end of a file
  - It indicates the file's contents end
- Function to detect (EOF) marker
  - `eof(internalFileName)`
    - True, if at end of file
    - False, otherwise



10.2 Using Loops to Process Files

Loops can be used to enter large amounts of data

```
For counter = 1 To numDays
    Display “Enter the sales for day #”, counter
    Input sales
    Write salesFile sales   // writes to the file
End For
```

Loops can also be used to read large amounts of data

```
While NOT eof(salesFile)
    Read salesFile sales
    Display currencyFormat(sales)
End While
```

10.3 Using Files and Arrays

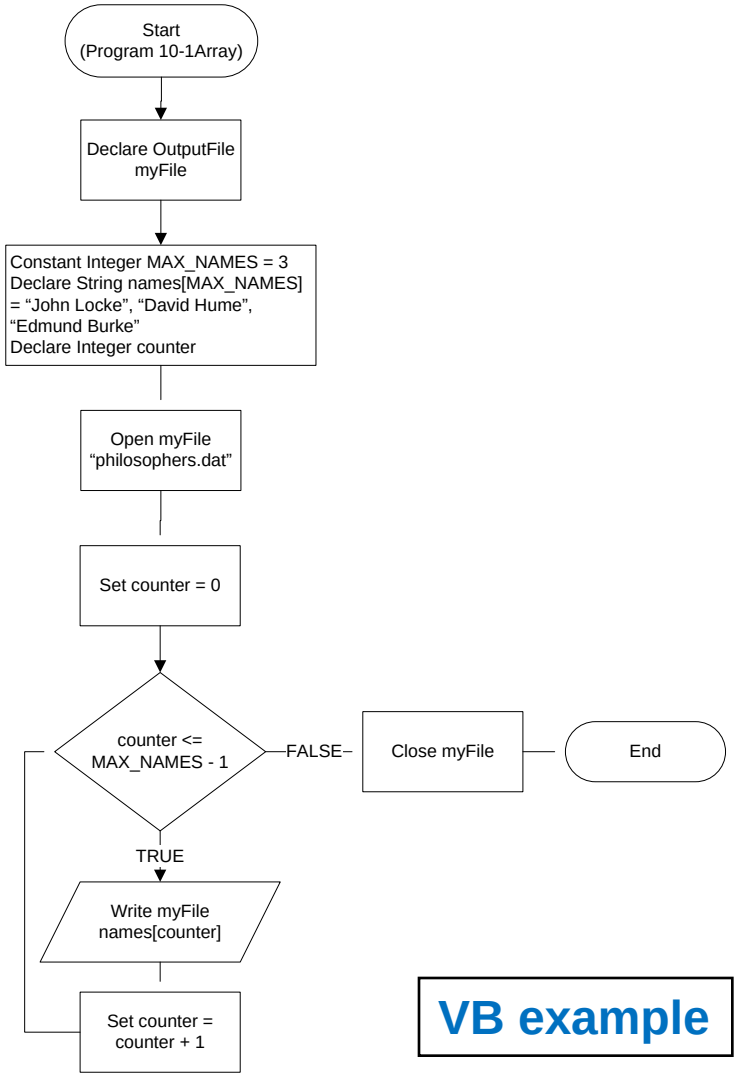
Files and arrays can be used together

- The contents of an array can be saved (written) to a file
  - Open the file
  - Use a loop to step through each element of the array
  - Write the contents to a file on each iteration
  - Close the file
- Modified Program 10-1
  - Writing philosopher names from array to file

10.3 Using Files and Arrays

Modified Program 10-1

```
Declare OutputFile myFile
Constant Integer MAX_NAMES = 3
Declare String names[MAX_NAMES] =
     “John Locke”, “David Hume”,
     “Edmund Burke”
Declare Integer counter
Open myFile “philosophers.dat”
For counter = 0 to MAX_NAMES – 1
     Write myFile names[counter]
End For
Close myFile
```



VB example

10.3 Using Files and Arrays

Files and arrays can be used together

- The contents of a file can be read into an array
  - Open the file
  - Use a loop to read each item from the file
  - Store each item in an array element
  - Close the file
- Modified Program 10-2
  - Reading philosopher names from file into array

10.3 Using Files and Arrays

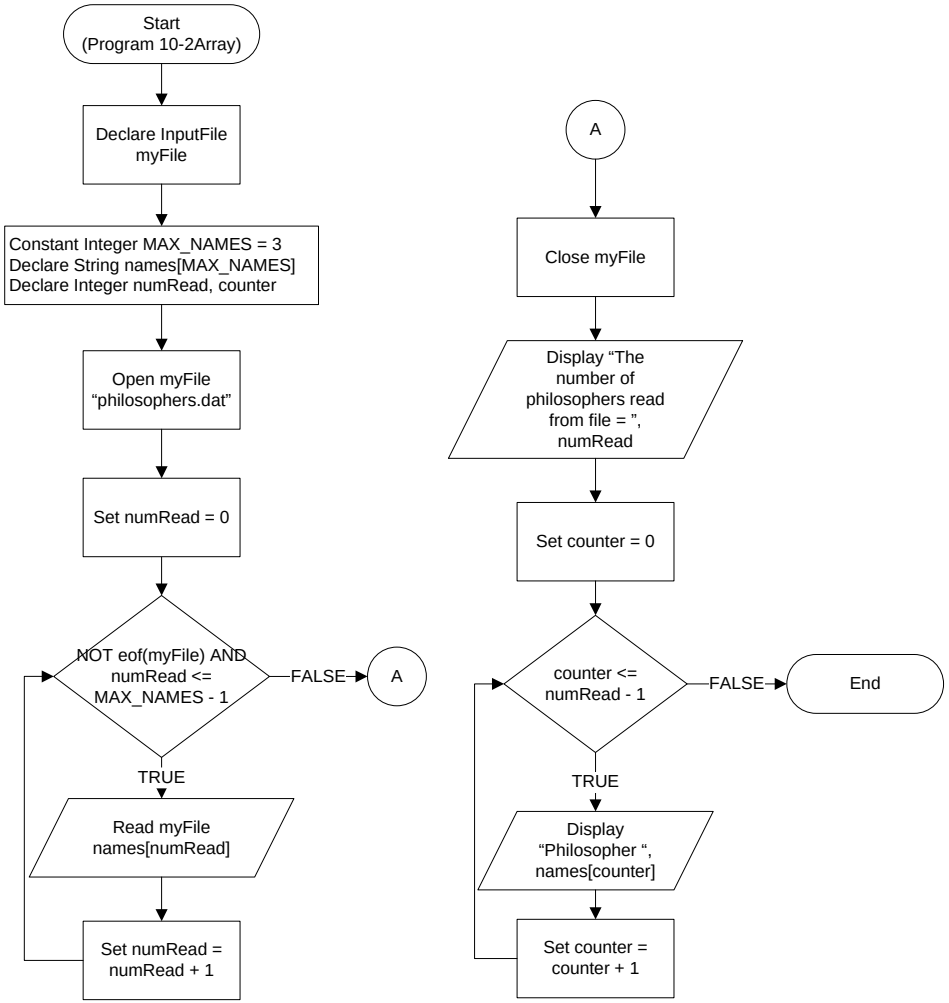
Modified Program 10-2 (pseudocode)

```
Declare InputFile myFile
Constant Integer MAX_NAMES = 3
Declare String names[MAX_NAMES]
Declare Integer numRead, counter
Open myFile “philosophers.dat”
Set numRead = 0
While NOT eof(myFile) AND numRead <= MAX_NAMES - 1
     Read myFile names[numRead]
     Set numRead = numRead + 1
End While
Close myFile
Display “The number of philosophers read from file = “, numRead
For counter = 0 to numRead – 1
     Display “Philosopher “, names[counter]
End For
```

VB example

10.3 Using Files and Arrays

Modified Program 10-2 (flowchart)



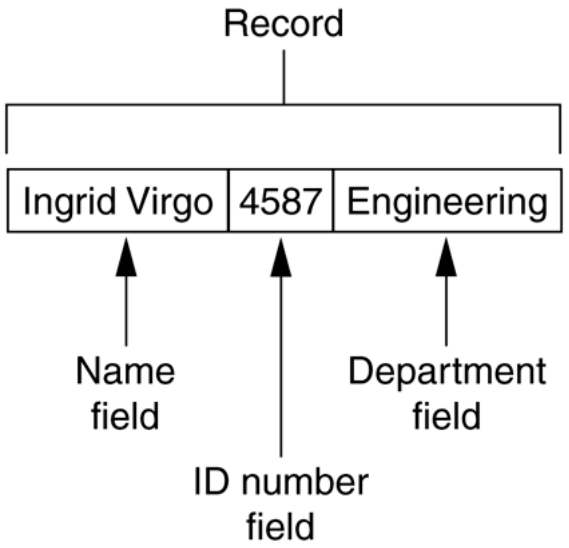
VB example

10.4 Processing Records

Data that is stored in a file is frequently organized in records

- A record is a complete set of data about an item
- A field is a single piece of data within a record

Figure 10-18 Fields in a record



10.4 Processing Records

Writing Records

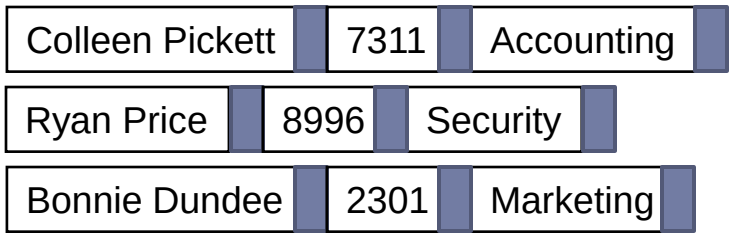
- An entire record is written using a single Write statement

```
Write employeeFile name, idNumber, department
```

Reading Records

- An entire record is read using a single Read statement

```
Read employeeFile name, idNumber, department
```



10.4 Processing Records

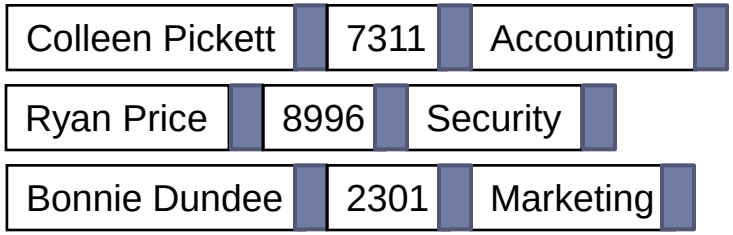
Writing Records Example: Program 10-7 *(pseudocode)*

```
Declare String name
Declare Integer idNumber
Declare String department
Declare Integer numEmployees, counter
Declare OutputFile employeeFile
Display “How many employee records do you want to create?”
Input numEmployees
Open employeeFile “employees.dat”
```

10.4 Processing Records

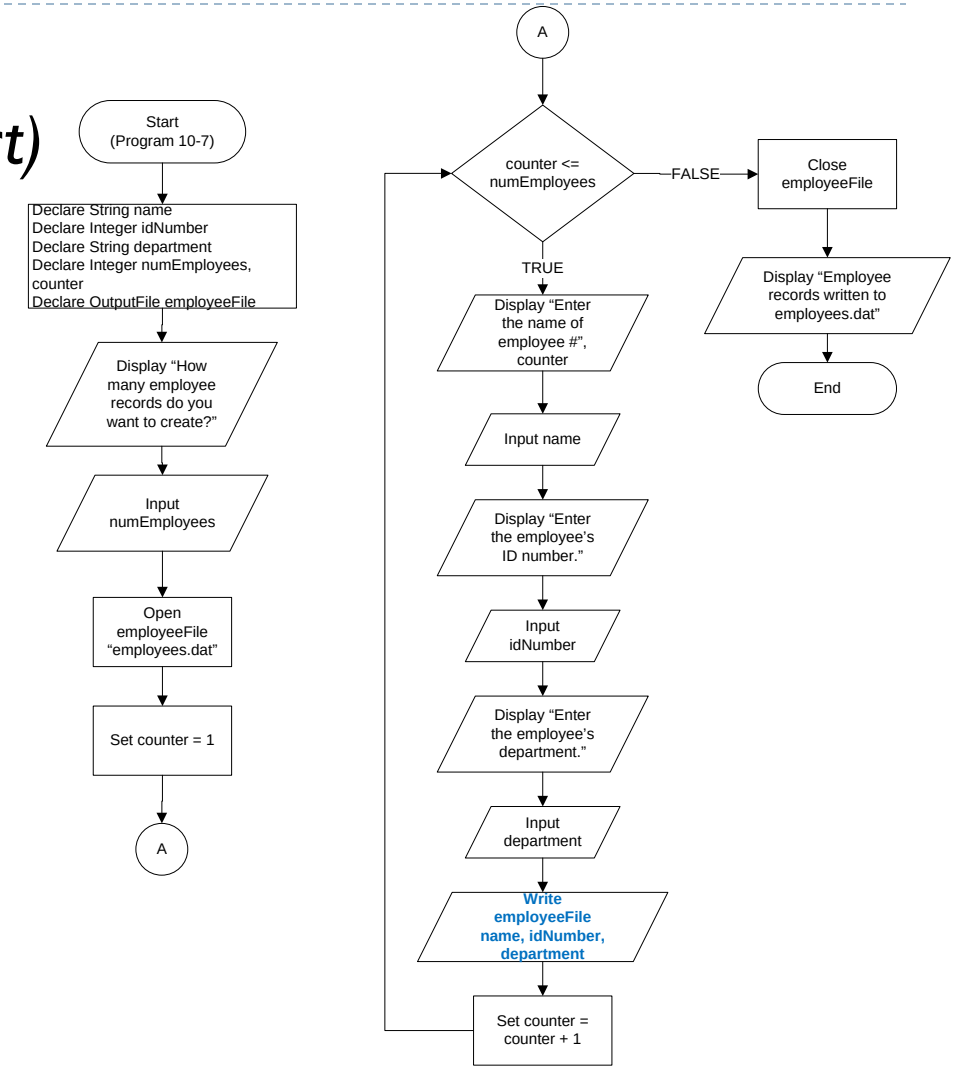
Writing Records Example: Program 10-7 *(pseudocode cont'd)*

```
For counter = 1 to numEmployees
   Display "Enter the name of employee #", counter
   Input name
   Display “Enter the employee’s ID number.”
   Input idNumber
   Display "Enter the employee's department."
   Input department
   Write employeeFile name, idNumber, department
   Display
End For
Close employeeFile
Display "Employee records written “
   "to employees.dat."
```



10.4 Processing Records

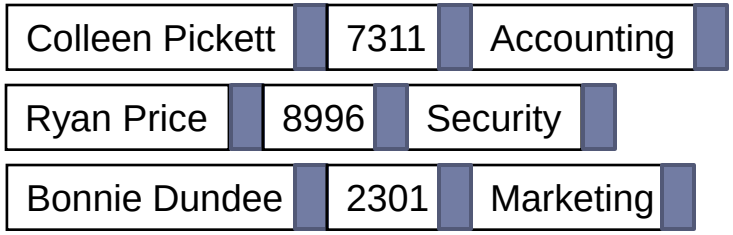
Writing Records Example: Program 10-7 *(flowchart)*



10.4 Processing Records

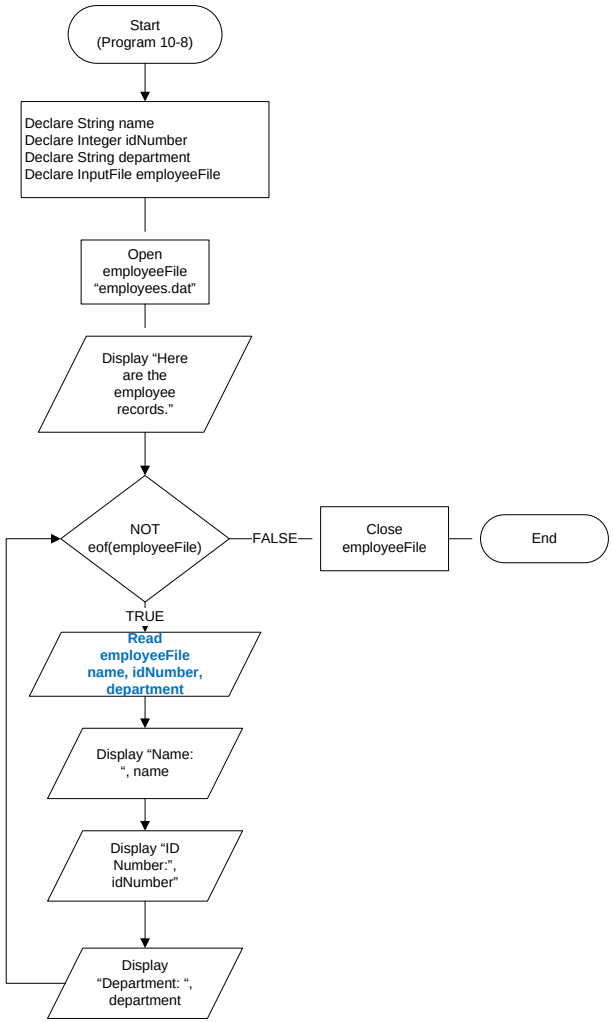
Reading Records Example: Program 10-8 *(pseudocode)*

```
Declare String name
Declare Integer idNumber
Declare String department
Declare InputFile employeeFile
Open employeeFile "employees.dat"
Display "Here are the employee records."
While NOT eof(employeeFile)
   Read employeeFile name, idNumber, department
   Display "Name: ", name
   Display "ID Number: ", idNumber
   Display "Department: ", department
   Display
End For
Close employeeFile
```



10.4 Processing Records

Reading Records Example: Program 10-8 *(flowchart)*



10.4 Processing Records

Algorithms can also be used for adding records to a file, searching for a specific record(s), modifying a record, and/or deleting a record

- Adding Records → Program 10-9
- Displaying all Records → Program 10-10
- Searching for a Record → Program 10-11
- Modifying a Record → Program 10-12
- Deleting a Record → Program 10-13

10.5 Control Break Logic

Control break logic interrupts (breaks) a program's regular processing to perform a different action when a control variable's value changes or the variable acquires a specific value

- Examples:
  - Book sales by city and state
  - Employees listed in order by department number
  - Charge account items grouped by category (Foods, Retail, Auto, etc.)
  - Line counter to pause output that goes beyond boundaries of screen output display

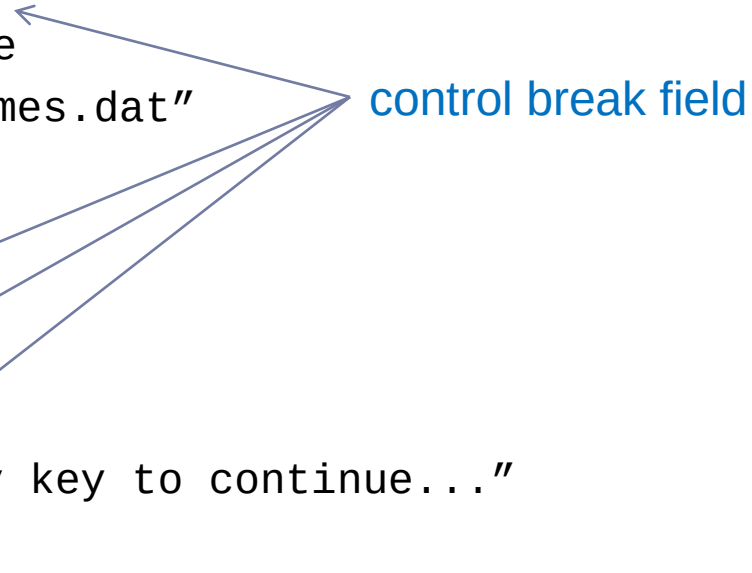
BOOK SALES BY CITY AND STATE

Ames	200
Des Moines	814
Iowa City	291
Total for IA	1305
Chicago	1093
Crystal Lake	564
McHenry	213
Springfield	365
Total for IL	2235
Springfield	289
Worcester	100
Total for MA	389
Grand Total	3929

10.5 Control Break Logic

Control Break Example: Program 10-14 *(pseudocode)*

```
Declare String name
Declare Integer lines = 0
Declare InputFile nameFile
Open nameFile "student_names.dat"
While NOT eof(nameFile)
   Read nameFile name
   Display name
   Set lines = lines + 1
   If lines == 24 Then
        Display "Press any key to continue..."
        Input
        Set lines = 0
   End If
End While
Close nameFile
```



10.5 Control Break Logic

Control Break Example: Program 10-14 *(flowchart)*

Figure 10-27 Pausing output after 24 items are displayed