CIS 141

# COVER PAGE

INTRODUCTION TO PROGRAMMING

Dondi Hanson

Zach Foutz

UNIT 6 EXERCISE



# TABLE OF CONTENTS

[COVER PAGE 1](#_Toc101335479)

[TABLE OF CONTENTS 2](#_Toc101335480)

[NARRATIVE 3](#_Toc101335481)

[DEFINING DIAGRAHM 4](#_Toc101335482)

[HIERARCHY CHART 5](#_Toc101335483)

[NASSI-SCHNEIDERMAN 6](#_Toc101335484)

[PROGRAM OUTPUT 7](#_Toc101335485)

[SOURCE CODE 8](#_Toc101335486)

[DESK CHECK 9](#_Toc101335487)

# NARRATIVE

The only problem I ran into this week was my system deciding to include the end of file into my array of records. I am not entirely sure if this is desired behavior, but some googling didn’t really turn up anything and my file is built correctly. Either way it’s just good practice to check for an EOF so that’s what I did and unset it from my array when I encountered it.

Otherwise all was good.

# DEFINING DIAGRAM

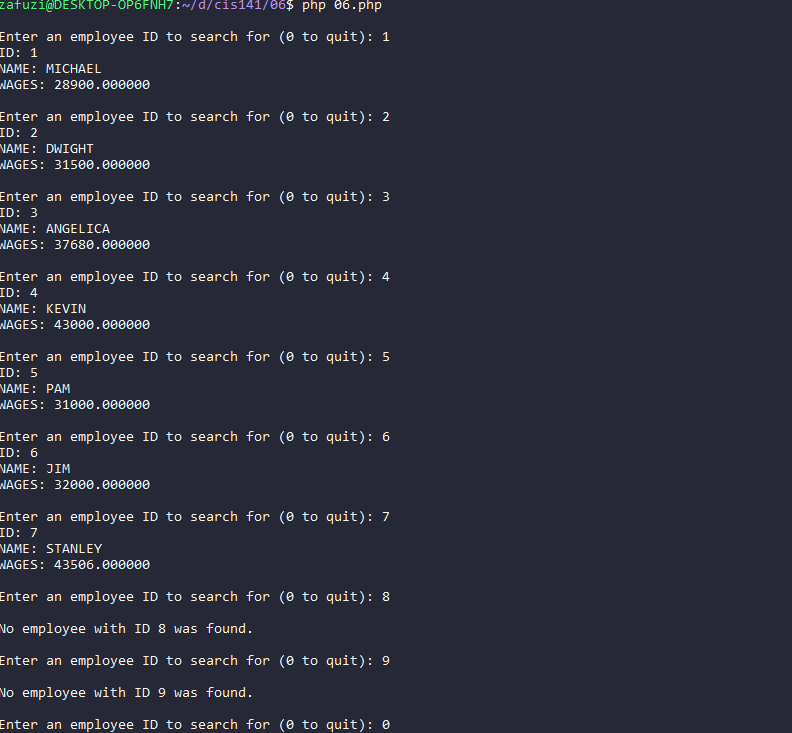
|  |  |  |
| --- | --- | --- |
| **INPUT** | **PROCESS** | **OUTPUT** |
| ID | OPEN EMPLOYEE RECORDS FILE | Employee record if found |
| NAME | Create employees array | Notice of none found if none found |
| WAGES | Set counter to zero |  |
|  | While not end of file |  |
|  | Read line from file into employees array |  |
|  | Make sure line is not empty |  |
|  | Unset array at counter if EOF |  |
|  | Increment counter |  |
|  | Repeat until end of file |  |
|  | Close file |  |
|  | Set e\_id to empty string |  |
|  | While e\_id is not “0” |  |
|  | Get ID from STDIN |  |
|  | Check employee array for ID |  |
|  | Print result |  |

# HIERARCHY CHART

(not required until unit 6 or 7)

NASSI-SCHNEIDERMA

# PROGRAM OUTPUT



# SOURCE CODE

<?php $input\_file = fopen("./employees.txt", "r") or die("Unable to read input file\n"); $employees = array(); $r = 0; while( !feof($input\_file) ) { $l = fscanf($input\_file, "%s%s%f", $employees[$r][], $employees[$r][], $employees[$r][]); if( $l == 0 ) { // remove from employees array if eof unset($employees[$r]); } $r++; } fclose($input\_file); $e\_id = ""; while( $e\_id != "0" ) { printf("\nEnter an employee ID to search for (0 to quit): "); fscanf(STDIN, "%s", $e\_id); $employee\_found = False; foreach( $employees as $e ) { if( strtolower($e[0]) == strtolower($e\_id) ) { // make all lowercase for easier testing printf("ID: %s\nNAME: %s\nWAGES: %f\n", $e[0], $e[1], $e[2]); $employee\_found = True; } } if(!$employee\_found && $e\_id != "0") { printf("\nNo employee with ID %s was found.\n", $e\_id); } } ?>



# DESK CHECK

Incoming Data: employees.txt

6 JIM 32000 2 DWIGHT 31500 4 KEVIN 43000

EOF

Where r is the record count and L is the count of found items on a line (normal is 3 for the input data)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| L# | Employees | R | E\_id | Employee\_found | L |
| 5 | [] |  |  |  |  |
| 6 |  | 0 |  |  |  |
| 8 | [[“6”, “JIM”, 32000]] |  |  |  | 3 |
| 12 |  | 1 |  |  |  |
| 8 | [  [“6”, “JIM”, 32000],  [“2”, “DWIGHT”, 32500]  ] |  |  |  | 3 |
| 12 |  | 2 |  |  |  |
| 8 | [  [“6”, “JIM”, 32000],  [“2”, “DWIGHT”, 32500],  [“4”, “KEVIN”, 43000]  ] |  |  |  | 3 |
| 12 |  | 3 |  |  |  |
| 8 | [  [“6”, “JIM”, 32000],  [“2”, “DWIGHT”, 32500],  [“4”, “KEVIN”, 43000],  [0]  ] |  |  |  | 0 |
| 10 | [  [“6”, “JIM”, 32000],  [“2”, “DWIGHT”, 32500],  [“4”, “KEVIN”, 43000]  ] |  |  |  |  |
| 12 |  | 4 |  |  |  |
| 16 |  |  | “” |  |  |
| 19 |  |  | “1” |  |  |
| 20 |  |  |  | False |  |
| 19 |  |  | “6” |  |  |
| 20 |  |  |  | False |  |
| 24 |  |  |  | True |  |
| 19 |  |  | “2” |  |  |
| 20 |  |  |  | False |  |
| 24 |  |  |  | True |  |
| 19 |  |  | “4” |  |  |
| 20 |  |  |  | False |  |
| 24 |  |  |  | True |  |

Output:

18 Enter an employee ID to search for (0 to quit): 1

28 No employee with ID 1 was found.

18 Enter an employee ID to search for (0 to quit): 6

23 ID: 6

NAME: JIM

WAGES: 32000

18 Enter an employee ID to search for (0 to quit): 2

23 ID: 2

NAME: DWIGHT

WAGES: 31500

18 Enter an employee ID to search for (0 to quit): 4

23 ID: 4

NAME: KEVIN

WAGES: 43000

18 Enter an employee ID to search for (0 to quit): 0

LOGIC:

7 while !feof true

9 3 == 0 false

7 while !feof true

9 3 == 0 false

7 while !feof true

9 3 == 0 false

7 while !feof false

9 0 == 0 true

17 while “” != “0” True

21 foreach($employees as $employee) (3)

22 1 == 6 false

22 1 == 2 false

22 1 == 4 false

27 if true && “1” != “0” true

17 while “1” != “0” True

21 foreach($employees as $employee) (3)

22 6 == 6 True

27 if False && “6” != “0” False

17 while “6” != “0” True

21 foreach($employees as $employee) (3)

22 4 == 6 False

22 4 == 4 True

27 if False && “4” != “0” False

17 while “4” != “0” True

21 foreach($employees as $employee) (3)

22 2 == 6 False

22 2 == 4 False

22 2 == 2 True

27 if False && “2” != “0” False

17 while “2” != “0” True

21 foreach($employees as $employee) (3)

22 0 == 6 False

22 0 == 4 False

22 0 == 2 False

27 if True && “0” != “0” False

17 while “0” != “0” False