San José State University
Department of Computer Science

CS 153
Concepts of Compiler Design
Fall 2017
Instructor: Ron Mak

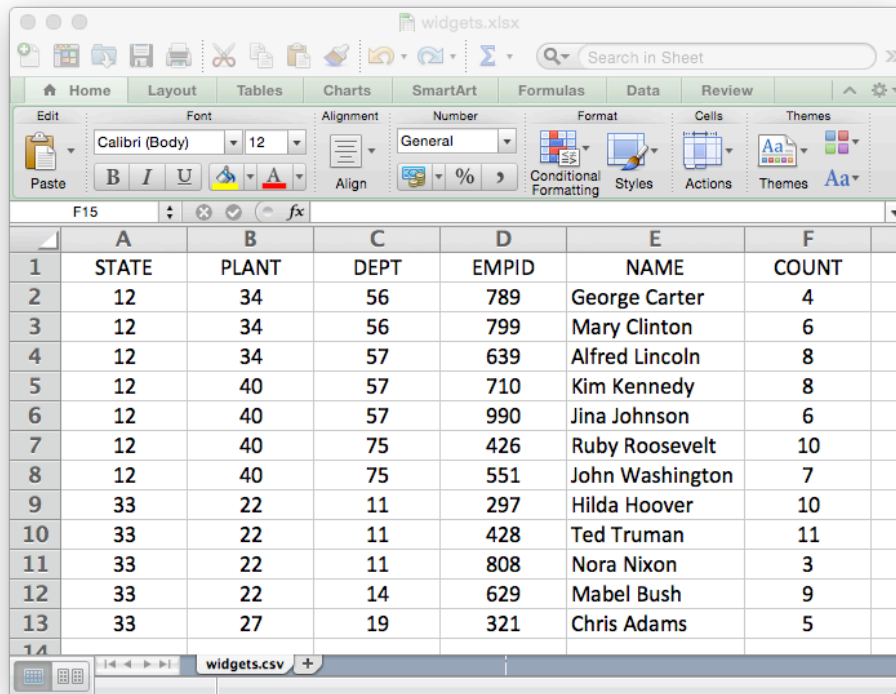## Assignment #1

Assigned:	Thursday, August 24
Due:	Friday, September 1 at 11:59 pm
	Individual assignment, 100 points max

Write a Pascal program

The purpose of this assignment is to give you a crash course in Pascal programming.

CSV files

CSV is a text file format saved from a spreadsheet. Here's the Excel spreadsheet **widgets.xlsx** that is the source of the CSV file that we'll use:

After the headers row, each row contains detail information about an employee: state code, plant (factory) code, department code, employee ID, name, and the count of widgets the employee made. The plants in a state are separate from the plants of another state, and the departments in a plant are separate from the departments of another plant. (Department 57 in Plant 34 is separate from Department 57 in Plant 40.)

The detail rows are already sorted first by state, then by plant, then by department, and finally by employee id.

Here's the CSV text file **widgets.csv** saved from the spreadsheet:

```
STATE,PLANT,DEPT,EMPID,NAME,COUNT
12,34,56,789,George Carter,4
12,34,56,799,Mary Clinton,6
12,34,57,639,Alfred Lincoln,8
12,40,57,710,Kim Kennedy,8
12,40,57,990,Jina Johnson,6
12,40,75,426,Ruby Roosevelt,10
12,40,75,551,John Washington,7
33,22,11,297,Hilda Hoover,10
33,22,11,428,Ted Truman,11
33,22,11,808,Nora Nixon,3
33,22,14,629,Mabel Bush,9
33,27,19,321,Chris Adams,5
```

You may assume that the file does not contain errors.

widgets.xlsx: http://www.cs.sjsu.edu/~mak/CMPE152/assignments/1/widgets.xlsx
widgets.csv: http://www.cs.sjsu.edu/~mak/CMPE152/assignments/1/widgets.csv

Expected output

Write a Pascal program **widgets.pas** that reads the CSV file and generates a detail report that contains **multi-level totals** to the standard output (i.e., the display):

```
STATE PLANT DEPT EMPID COUNT NAME

  12    34    56   789      4 George Carter
  12    34    56   799      6 Mary Clinton

                           10 TOTAL FOR DEPT   56 *

  12    34    57   639      8 Alfred Lincoln

                            8 TOTAL FOR DEPT   57 *
                           18 TOTAL FOR PLANT 34 **

  12    40    57   710      8 Kim Kennedy
  12    40    57   990      6 Jina Johnson

                           14 TOTAL FOR DEPT   57 *

  12    40    75   426     10 Ruby Roosevelt
  12    40    75   551      7 John Washington

                           17 TOTAL FOR DEPT   75 *
                           31 TOTAL FOR PLANT 40 **
                           49 TOTAL FOR STATE 12 ***

  33    22    11   297     10 Hilda Hoover
  33    22    11   428     11 Ted Truman
  33    22    11   808      3 Nora Nixon

                           24 TOTAL FOR DEPT   11 *

  33    22    14   629      9 Mabel Bush

                            9 TOTAL FOR DEPT   14 *
                           33 TOTAL FOR PLANT 22 **

  33    27    19   321      5 Chris Adams

                            5 TOTAL FOR DEPT   19 *
                            5 TOTAL FOR PLANT 27 **
                           38 TOTAL FOR STATE 33 ***

                           87 GRAND TOTAL         ****
```

The report shows the total widgets made by each department, plant, and state, and a grand total.

Your output should match the sample output with regards to field spacing and blank lines. The stars indicate the levels of totals.

widgets.out: http://www.cs.sjsu.edu/~mak/CMPE152/assignments/1/widgets.out

Pascal program requirements

Your Pascal program must have the following features:

- A record data type to contain the information for each employee.
- Nested procedures and functions, e.g., a procedure nested inside a procedure.
- At least one parameter passed by reference.
- **Do not use the built-in string type.** Represent strings as arrays of characters, as in classic Pascal. For example:

  ```
  TYPE
      Field = PACKED ARRAY [1..12] OF char;
  ```

  (Ignore the `PACKED` reserved word.)

Text input hints

- If `ch` is a character variable, then `read(ch)` will read the next character from the standard input.
- A call to `readln` will skip the rest of the current input line, and the next call to `read` will read starting on the next input line.
- Boolean functions `eoln` and `eof` without parameters test whether the standard input is at the end of the current line or at the end of file, respectively. (If you're typing input data into a Windows command window, a line containing only control-Z is the end-of-file marker. On Linux and the Mac, it's control-D.)

Your program, including comments and blank lines, shouldn't be over 300 lines. Even though Pascal is a simple language compared to Java, learning a new language can be tricky. **Do not wait until the last minute** to work on this assignment!

What to submit to Canvas

- Your reasonably commented Pascal source file **widgets.pas**, as a text file.
- A text file of your program's output report.

Zip the two files together and submit the zip file into Canvas: **Assignment #1**

> **This is an individual assignment. Your program must be your own work.**

Rubric

Your program will be graded according to these criteria:

Criteria	Maximum points
Good output	**40**
• Correct output values.	• 20
• Correct report format.	• 20
Good program design	**30**
• Record data type.	• 10
• Nested procedures and/or functions.	• 10
• At least one parameter passed by reference.	• 10
Good program style	**30**
• Descriptive variable names.	• 10
• Meaningful comments.	• 10
• Nicely indented code.	• 10

A quick tour of the Lazarus Pascal IDE

A very good IDE (integrated development environment) you can use to develop Pascal program is Lazarus, which you can download for Linux, Mac, or Windows: https://www.lazarus-ide.org/index.php

Download and install Lazarus. After you start it for the first time, close all of its windows except for the main menu bar. Select the menu item File | New. In the popup window, select Program (under Project) and click the OK button. An editor window with the name project1.lpr will open. Delete all the default text. Select the menu item File | Save As and choose the name **widgets.pas**.

Now you can edit your program and save it periodically. To compile, select the menu item Run | Build. If there are no syntax errors, Lazarus will generate an executable file (Widgets.exe on Windows) which you can run on the command line as shown above.

You can also run your program from within Lazarus. Select the menu item Run | Run. A console window will open into which you can type (or paste) the input data. There are advantages of running within Lazarus: you can set breakpoints, look at the values of local variables, get the runtime stack, etc. Unfortunately, the console window (which contains the program output) disappears immediately when your program finishes.

A very useful link: Pascal runtime error codes: https://www.freepascal.org/docs-html/user/userap4.html