

CSE 1301 PRACTICAL SESSION 9

Structs (10 marks)

Aim: by the end of this practical session, you should be able to write programs which use structs: create a struct type, store and retrieve information from struct members, pass a struct variable to functions, return structs from functions, etc.

Coding style and documentation: You are expected to document your programs and to use a standard, clear, and consistent coding style. Up to 2 of the marks for each practical class may be deducted for poor coding style and/or inadequate documentation.

Preparation (to be completed before class) (2 marks)

Write all the algorithms required for the questions below, and attempt the code for all the questions.

PART 1: Structs (3 marks)

- (a) Declare a structure type, Order, that is suitable for containing information about the orders in a florist's shop. The structure should contain members for the type of flower (a string), the delivery time (an integer), the price (a float), and the customer's name (a string).
- (b) Write a function that will read in values for each member of an Order from the keyboard (with suitable prompts for the user), and return the completed struct.
- (c) Write another function that prints out the values in an Order.
- (d) Write a main program to test your functions. Save your program as Prac9-1d.c
- (e) Now extend the structure such that a customer's name is itself a structure consisting of a first name (string) and a last name (string). This means that the Order structure is now a nested structure. Revise your program from part (b-d) to handle this extension. Save your revised program as Prac9-1e.c

PART 2: Array of Structs (5 marks)

- (a) Alter your main program from Part 1 so that it contains an array of Orders, rather than just a single order. Additionally, create a simple menu of options to allow the user to perform the tasks outlined in the following parts (b) to (e).
- (b) The first menu option will allow the user to add a new order to the array. For this the program will need to call the read function and store the resulting struct in the next available element of the Orders array. The program will also need to maintain the position of the next available element.
- (c) The second menu option will allow the user to display the current list of orders. This is to be done by altering your output function (from Part 1 (c)) to print out an array of Orders, rather than just a single Order.

- (d) The third menu option will allow the user to display the next job to be worked on. This is accomplished by extending your structure with an additional member (Boolean) to represent whether a job has been delivered yet. Write a function that can be used to select the next job to be worked on. This function takes the struct array as an argument and returns the job with the earliest delivery time (smallest integer) that has not been marked as delivered. The function marks this job as delivered and prints out its complete information using the function from Part 1 (c).

Make sure you test all parts of your program thoroughly.

Submission:

Preparation: Show written preparation work to demonstrator at start of class for marking. If not done before class, it will not receive any marks, however you should still show it to your demonstrator during class to check your understanding. You must submit your program to the cheater checker, which is available on the web via the CSE1301 courseware page.

PART 1 and 2: To be marked by the demonstrator during this class. No late submission.

Additional Component

(2 BONUS MARKS)

- (a) Include a menu option in your program to delete an order from the Order array. You will need to ask the user which order they wish to delete and check that they have made a valid selection. You will also need to deallocate the element of the array in which the deleted order was stored.
- (b) Add a menu option that allows the user to print the information for all undelivered orders in increasing order of delivery times. This can be achieved using a Boolean member of your structure to record whether or not the entry has been displayed. This Boolean member is initialised to FALSE before display commences. Write a similar function to Part 2 (d) that selects the earliest delivery time that has not been marked as delivered or displayed. Print out this job and mark it as being displayed. You will need to make a number of calls to this function in order to display all undelivered orders.