

CSE 130 Fall 2009

Programming Assignment 5

Due Date: 11am, Friday Nov 13, 2009. Use `~cse130/submit` command from your sparky account. Do not forget to change to your working directory first. Include your name and ID number as a comment. Make sure that it runs on sparky. Write comments in your code.

Objective: This assignment will help you get some experience with structures and nested function calls. It just looks complicated, but it is straight forward and not much code.

Outline: Write a C program that asks user to type in two amounts in dollars and cents. It then calls several functions and calculates their sum, difference, selects the larger amount, converts them to dollars (decimal, float) and cents (integer), and rounds them to nearest dollar amount (integer). See the example given at the end.

Technique: Declare a global structure for type amount as shown in the program structure. Declare two variables a, b of type amount as shown. Read a.dollars, a.cents, b.dollars and b.cents as supplied by the user. Write several functions, one each for a specific task. Call these functions in the order given and print results. ALL PRINTING IS TO BE DONE ONLY FROM main ().

Functions: In addition to main (), you MUST write these six functions. You may write additional functions if you need.

(1) A function **add(amount x, amount y)** for adding two amounts x and y. It returns their sum as a structure of type amount. Add cents first. If more than 100, subtract 100, and adjust dollar sum by 1.

(2) A function **larger(amount x, amount y)** for selecting the larger of the two amounts. If x is larger it returns 1, else it returns 0.

(3) A function **difference(amount x, amount y)** that subtracts the smaller amount from the larger one and returns it as a structure. It first calls function larger and decides how the subtraction is to be performed (nested function call).

(4) A function **allDollars(amount x)** converts the amount x stored as a structure to a floating point value, and returns that value. For example if x is 5 dollars and 21 cents, it returns 5.21 as a floating point value (casting may be required).

(5) A function **allCents(amount x)** converts the amount x stored as a structure to cents (an integer value), and returns that value. For example if x is 5 dollars and 21 cents, it returns 521 as an integer value.

(6) A function **round(amount x)** to round the amount x to its nearest dollar-value. It returns this value as an integer. Add 50 to x.cents and decide if you want to add 1 to x.dollars. Return the dollar-only value as an integer.

Input: First ask for two amounts as shown. Read amount1 using scanf, with two integer values for dollars and cents typed on the same line. Next, read amount2 using code similar to that for amount1.

Output: ALL PRINTING IS TO BE DONE ONLY FROM main (). First, add amount1 and amount2 by calling add function. Print their sum as shown. Next, call function larger and print the larger of these two amounts. After this, call function difference to get their difference and print that. NOTE THAT function difference WILL CALL function larger AGAIN. Then print amount1 as a

floating point value by calling function allDollars. Call function allCents and print amount1 as cents (integer value). Finally, call round function to round out both amount1 and amount2. Print them as integer values.

Example Session: Here is an example session (on sparky).

Type in amount-1 as dollars and cents.

10 76

Type in amount-2 as dollars and cents.

42 31

Sum of two amounts is: dollars 53 and cents 7

Larger of the two amounts is: dollars 42 and cents 31

Their difference is: dollars 31 and cents 55

Amount-1 in dollars is: dollars 10.76

Amount-1 in cents is: cents 1076

Rounded amount-1 is: dollars 11

Rounded amount-2 is: dollars 42

Do not use any **global variables**. Use a global declaration for type amount as shown on the next page.

Program Structure is on the next page.

Program Structure:

```
#include <stdio.h>

// Global declaration for type amount.

struct amnt // struct amnt will be redefined as amount.
{
    int dollars;
    int cents;
};

typedef struct amnt amount; // This defines amount.

// amount is another name for type struct amnt.
// Now type amount is available to all functions.

// Code for function add

// Code for function larger

// Code for function difference

// Code for function allDollars

// Code for function allCents

// Code for function round. You will get a warning
// if you used round. It is a built-in function.

// Function main begins here.

int main ()
{
    // Declare structure variables a, b for
    amount1 and amount2.
    // Additional Variable declarations .

    printf("Type in amount-1 as dollars and cents.\n");
    scanf ("%d%d", &a.dollars, &a.cents );

    printf("Type in amount-2 as dollars and cents.\n");
    scanf ("%d%d", &b.dollars, &b.cents );

    // Call functions as required and print results.

    return 0;
}
```