Advanced Programming Techniques (a.k.a. Programming in ANSI / ISO C)

Module 2

Tutorial Questions

Objectives

This tutorial provides practice with defining and using structs. Although discouraged in future tutorials, this tute allows use of 'magic numbers'.

Activities

1. Declare in a header file a struct called `card` for storing the following information about a playing card:
   - o the 'pips' - an integer (where 1 = ace, 2..10, 11 = jack, 12 = queen, 13 = king)
   - o the 'suit' - a char (‘H’ = heads, ‘C’ = clubs, ‘S’ = spades, ‘D’ = 'diamonds')

   In the main program, declare a variable of this struct, and assign the card the values of 2 of Hearts. Use `printf` to print out the pips and suit of the card.

2. Declare the card struct as a new type: `Card`. Change your main program to use this new data type.

3. Write a new void function `display` which accepts a `Card` as a parameter and prints out the pips and suit of the card. The function should be written *under* the main program block. The function is called from main and passed the main function's local Card variable. Does the program compile? Add a function prototype for display to the header file. What effect did this have?

4. Define a new type which is the enumeration: *clubs, diamonds, hearts, spades*. Alter the definition of `Card` so the suit is of this type. Alter function `display` to use a `switch` statement, to print "Clubs" if the suit is of enumeration *clubs* etc.

5. Change the main function’s local `Card` variable to an array of 4 `Card`s, called `hand`. Assign some pips and suit values to each card in the array. Use a `for` loop to iterate over the array, passing the array element to function `display`.

6. Define a new datatype called `Hand` which is an array of 4 `Card`. Alter the main local variable, `hand`, to be of this type. Write a void function called `setHand` which accepts a parameter of type `Hand`, and assigns values to each element.