**Project:** **p04**

**Assignment:** Program **p04** makes change. Given an amount, program **p04** finds the number of bills and the denomination of each bill needed to sum to the dollar value of the amount. Program **p04** also finds the coins required summing to that portion of the amount less than one dollar.

Program **p04** prompts the user for an amount. If the amount is greater than $12,000, program **p04** does not make change and prompts the user again. If the amount is between one cent and $12,000, program **p04** makes change. If the amount is less that one cent program **p04** terminates. Program **p04** continues to prompt the user for amounts until an amount less than one cent is entered.

Several examples are shown in Figure 1. User input is italicized and marked in bold.

Please use the following denominations for currency in bills, $10,000, $5,000, $1,000, $500, $100, $50, $20, $10, $5, and $2.

Please use the following denominations for coins, one-dollar, one-half-dollar, quarter, dime, nickel, and penny.

You must employ functions to receive full credit for this project.

**Program Files:** Project **4** consists of file **p04.cpp**.
File **p04.cpp** must be stored in the root directory of your student account. Failure to store project files in the root directory of your student account will result in a score of ***zero*** (***0***) for this project.

**Command Line:** Executable file **p04** is invoked with no program parameters as shown below.

```
$ p04
$ p04
Enter the amount for which you want change. 934.16
1 five-hundred dollar bill
4 one-hundred dollar bills
1 twenty dollar bill
1 ten dollar bill
2 two dollar bills
1 dime
1 nickel
1 penny

Enter the amount for which you want change. 1000.00
1 one-thousand dollar bill

Enter the amount for which you want change. 12000.00
The amount is too large. Try again.

Enter the amount for which you want change. 0.01
1 penny

Enter the amount for which you want change. 9999.99
1 five-thousand dollar bill
4 one-thousand dollar bills
1 five-hundred dollar bill
4 one-hundred dollar bills
1 fifty dollar bill
2 twenty dollar bills
1 five dollar bill
2 two dollar bills
1 half dollar
1 quarter
2 dimes
4 pennies

Enter the amount for which you want change. -1.00
$
```

**Figure 1.** Output format.

Algorithm:

**//**-----------------------------------------------------------------------------------------------------------------------

**//**Function *TheBill* prints the number of bills having the value stored in parameter *denomination*. Bills are
//taken from parameter *amount*. Function *TheBill* returns the amount remaining after bills have been
//removed. The spelling of the denomination is given in parameter *denominationspelling*. Example: if
//parameters *amount*=4976, *denomination*=1000, and *denominationspelling*=”one thousand”, function
//*TheBill* prints “4 one thousand dollar bills” and returns 976.

**//**-----------------------------------------------------------------------------------------------------------------------

*TheBill*

1. Function *TheBill* returns a value of type **int**.
2. Declare parameter *amount* having type **int**. Parameter *amount* is a value parameter. Bills having a denomination given in parameter *denomination* are taken from parameter *amount*.
3. Declare parameter *denomination* having type **int**. Parameter *denomination* is a value parameter. Parameter *denomination* contains the amount of the denomination.
4. Declare parameter *denominationspelling* having type *string*. Parameter *denominationspelling* is a reference parameter. If parameter *denomination* has a value of 500, parameter *denominationspelling* has a value of “five hundred”

5. Declare variable *quantity* having type **int**.
6. Initialize *quantity* to the number of bills having a denomination equal to the value stored in parameter *denomination* that are contained in parameter *amount*.
7. Declare variable *returnvalue* having type **int**.
8. Initialize *returnvalue* to the amount that remains after all bills equal to the denomination stored in parameter *denomination* are removed.
9. **if** the quantity of bills is greater than zero **then**
   - 9.1. Print *quantity*
   - 9.2. Print *denominationspelling*
   - 9.3. Print the string **“dollar bill”**
   - 9.4. if the quantity of bills is greater than one, print **“s”**
   - 9.5. For example, if *amount*=4976, *denomination*=1000, and *denominationspelling*=”one thousand”, print “4 one thousand dollar bills”
10. **return** *returnvalue*.

**//**-----------------------------------------------------------------------------------------------------------------------

**//**Function Bills directs the process of printing the quantity and denomination of each bill that sums to input
//parameter amount

**//**-----------------------------------------------------------------------------------------------------------------------

*Bills*

1. Function *Bills* returns type **void**.
2. Declare parameter *dollars* having type **int**. Parameter *dollars* is a value parameter.

3. Call function *TheBill* to print the number of ten thousand dollar bills in parameter *dollars*. Function *TheBill* returns the amount remaining after all ten thousand dollar bills have been removed. Parameter *dollar* is assigned the amount returned from function *TheBill*.
4. Call function *TheBill* to print the number of five thousand dollar bills in parameter *dollars*. Function *TheBill* returns the amount remaining after all five thousand dollar bills have been removed. Parameter *dollar* is assigned the amount returned from function *TheBill*.
5. Call function *TheBill* to print the number of one thousand dollar bills in parameter *dollars*. Function *TheBill* returns the amount remaining after all one thousand dollar bills have been removed. Parameter *dollar* is assigned the amount returned from function *TheBill*.
6. Call function *TheBill* to print the number of five hundred dollar bills in parameter *dollars*. Function *TheBill* returns the amount remaining after all five hundred dollar bills have been removed. Parameter *dollar* is assigned the amount returned from function *TheBill*.

7. Call function *TheBill* to print the number of one hundred dollar bills in parameter *dollars*. Function *TheBill* returns the amount remaining after all one hundred dollar bills have been removed. Parameter *dollar* is assigned the amount returned from function *TheBill*.
8. Call function *TheBill* to print the number of fifty-dollar bills in parameter *dollars*. Function *TheBill* returns the amount remaining after all fifty-dollar bills have been removed. Parameter *dollar* is assigned the amount returned from function *TheBill*.
9. Call function *TheBill* to print the number of twenty-dollar bills in parameter *dollars*. Function *TheBill* returns the amount remaining after all twenty-dollar bills have been removed. Parameter *dollar* is assigned the amount returned from function *TheBill*.
10. Call function *TheBill* to print the number of ten-dollar bills in parameter *dollars*. Function *TheBill* returns the amount remaining after all ten-dollar bills have been removed. Parameter *dollar* is assigned the amount returned from function *TheBill*.
11. Call function *TheBill* to print the number of five-dollar bills in parameter *dollars*. Function *TheBill* returns the amount remaining after all five-dollar bills have been removed. Parameter *dollar* is assigned the amount returned from function *TheBill*.
12. Call function *TheBill* to print the number of two-dollar bills in parameter *dollars*. Function *TheBill* returns the amount remaining after all two-dollar bills have been removed. Parameter *dollar* is assigned the amount returned from function *TheBill*.
13. Call function *TheBill* to print the number of one-dollar bills in parameter *dollars*. Function *TheBill* returns the amount remaining after all one-dollar bills have been removed. Parameter *dollar* is assigned the amount returned from function *TheBill*.

**int** *main***()**

```
1. Declare variable amount having type double. Variable amount is used to store the amount. Program p04 produces the minimum number of bills and coins that add up to the amount.
2. repeat
    2.1. Prompt the user. Print "Enter the amount for which you want change. "
    2.2. Read the amount from the keyboard.
    2.3. if the amount is greater than $10000.00 then
        2.3.1. Print "The amount is too large. Try again."
    2.4. else
        2.4.1. Declare variable dollars having type int.
        2.4.2. Assign the integer portion of amount in dollars.
        2.4.3. Call function Bills to print the bills that sum to the integer portion of amount.
        2.4.4. Declare variable cents having type int.
        2.4.5. Assign the integer portion of $100(amount - dollars)$ in cents.
        2.4.6. Call function Coins to print the coins that sum to the fractional portion of amount.
3. until amount < 0
```