

LAB 6: FUNCTIONS – PART 2

For each problem below:

- Analyze the problem by identifying input, output, formula, and constraint
- Design an algorithm to solve the problem using pseudocode (so that you could include the pseudocode in your program)
- Prepare several, appropriate number of test data to verify the correctness of your program
- Prepare, compile, link, and execute the program to solve the problem
- Test your program using the prepared test data
- Write proper documentation in the program. Include the following information to form a banner at the beginning of your program:

```

/*****
* Author's: your name and student ID
* Course: the course code only
* Section: your specific section number
* Date: of lab session
* Brief description: of what problem the
*                  program tries to solve
* Pseudocode: write the algorithm to solve the
*                  problem
* Test data: provide a set of test data
*            - input & expected output
*****/

```

QUESTIONS

- Given the `main()` function as in (a):

| | |
|---|---|
| <pre> void main(void) { char op; float x, y; printMenu(); x=getNumber(); y=getNumber(); op=getOperator(); doOperation(op, x, y); } </pre> | <pre> ----- Simple Calculator Program ----- A - Addition S - Subtraction M - Multiplication D - Division ----- </pre> |
|---|---|

(a)

(b)

Write a complete calculator program, which is implemented with several programmer-defined functions: The program begins with printing a menu as shown in (b) via the `printMenu()` function. Then call another function, named `getNumber()`, twice to obtain two real numbers from the user. Next, prompt and get an operator from the user via the third `getOperator()` function. Within the `getOperator()` function, validate the operator entered by the use and keep on prompting an invalid input message such as “Invalid operator. Please re-enter” as long as invalid operator is entered and ask the user to re-enter the operator. Once validated, return the operator to the `main()` function. Then, perform the operation to the two numbers based on the selected operator in a function called `doOperation()`.

- The National Earthquake Information Center has asked you to write a program implementing below decision table to characterize an earthquake based on the Richter Scale. The program

must implement multiple alternatives to display a message indicating the damage level based on its Richter scale number. Print a message to indicate a bad data as well. The information in the decision table shall be displayed as a menu and terminate the program only when the user enters -999.

| Ritcher Scale | Characteristics |
|---------------|--|
| 0 – 5.0 | Little or no damage |
| 5.1 – 6.5 | Serious damage or wall cracks |
| 6.6 – 7.5 | Disaster or houses and building collapse |
| Higher | Catastrophe, most building destroyed |

Your program shall be broken into two programmer-defined functions as follows:

1. `getInput()` function that reads the Richter scale value from the user. Prompt "Enter richter scale or -999 to quit:" message before you read the value.
 2. `printMsg()` function that displays the appropriate message or bad data message given the Richter scale
3. Write a program that print out the total score of carry mark and final exam mark and its associated message for certain number of students. The final exam mark forms 50% of the total score. For example, if the exam mark is 95 then the exam percentage is 47.50. The message to be printed on the screen must be based on the following:

| Total Score | Message |
|-------------|---|
| 80 to 100 | You are an extraordinary student. |
| 60 to 79 | You are good at this course. |
| 50 to 59 | Not good enough but don't give up easily. |
| 0 to 49 | You better repeat this course |

Apply data validation in this program. Prompt error messages and ask the user to repeat entering the data as long as invalid values are entered. The valid ranges are in between 0 to 50 for the carry mark and in between 0 to 100 for the exam mark. The program's input/output should be formatted by following the sample shown in Figure 1.

This program must be implemented with multiple programmer-defined functions as follows:

- `printMenu()` – prints the menu as shown below.
- `getCarryMark()` – reads carry mark from user and returns it the `main()` function.
- `getFinalMark()` – read final exam mark from user and returns it the `main()` function.
- `calculateScore()` – given both carry mark and final exam mark, calculates a student's score (based on abovementioned information) and returns the score to the `main()` function.
- `printMessage()` – given the score, prints on the screen the abovementioned message.

```
-----
      Printing Messages Just for Fun
-----
How many students? 2

Student: 1
Enter carried mark: 400.0
      Invalid data. Please re-enter.

Enter carried mark: 40.0
Enter final exam mark: 915.00
      Invalid data. Please re-enter.

Enter carried mark: 40.0
Enter final exam mark: 95.00
      Total score is 87.50

You are an extraordinary student.
-----

Student: 2
Enter carried mark: 32.0
Enter final exam mark: 88.00
      Total score is 76.00

You are good at this course.
-----

Have a nice day!
```

Figure 1: Sample input/output

4. You are required to write a complete C program that has several basic calculator functionalities, which must be implemented using modular programming concept. The program must be able to add, subtract, multiply and divide two values. User should be prompted to enter the operator option and two numbers. Then the numbers should be calculated based on the selected option. Finally the program should clearly display the operands, operator and result of the operation. The program's output must be formatted with two decimal places. The function main of the program should read the operator option and call three other user-defined functions as follows:
- `getInput()` - reads one value from user and return the value to the main function
 - `doOperation()` - performs the selected operation to the two values entered and returns the result of the operation to the main function
 - `printAnswer()` - display the result of the operation

The input/output of the program should look like the sample shown in (c) and should guide you in writing the proper code.

```
Enter an operator ( +, -, x, / ): +
Enter any number: 37.81476
Enter any number: 54.78345

Answer for 37.81 + 54.78 is 92.60
```

5. Many people buy gemstones by type -- they want to buy a sapphire or a tourmaline or an amethyst. But one thing XYZ company has learned in the colored gem business is that most customers are concerned above all with color. A list of some gem types organized by color and other information is as shown below:

Table 1: List of gemstones

| Code | Color | Gem Type | Price/gram (USD) |
|------|--------|------------|------------------|
| R | Red | Ruby | 100.00 |
| P | Pink | Sapphire | 120.00 |
| B | Blue | Aquamarine | 88.00 |
| V | Violet | Amethyst | 250.00 |

You are required to write a program with multiple functions that can be used to determine the color and the type of a gemstone as well as the amount due for each customer as follows:

- `printMenu()` – this function prints of the screen the list of gemstones as in Table 1.
- `getCode()` – this function asks the user to enter the code used to represent the gemstone, which is then returned to the function `main()`.
- `getQty()` – this function asks the user to quantity of gemstones purchased, which is then returned to the function `main()`.
- `printType()` – this function prints the gem type given the code information.
- `calculateAmtDue()` – this function calculates the price that the user has to pay, given the quantity, and code information. Return the price to the `main()` function before the value is printed on the screen from the `main()` function.

Before the program ends, prints “[Thank you. Please come again]”. The program’s input/output should be formatted by following the sample shown in Figure 2. Add proper data validation in this program. A sample input/output of the program with invalid data is shown in Figure 3.

```

~ Welcome ~

Enter code -> B
Enter quantity (in gram) -> 10.5

Gem Type is Aquamarine (Color is Blue)
Amount due: USD924.00

[Thank you. Please come again]
```

Figure 2: Sample input/output

Add the list of
gemstones here

```

~ Welcome ~

Enter code -> C
Enter quantity (in gram) -> 0

Unknown gem type!
Invalid quantity purchased!

[Thank you. Please come again]
```

Figure 3: Sample input/output with invalid data