

Homework #6

(Chapter 6)

Programming Requirements:

1. Name your project using your full name, the homework # and the problem #, and your class section.

e.g.: **John Smith-HW6.1-MW**

2. After you create the framework of your program, you should first create an algorithm as a list of instructions that shows how to solve the problem step by step. Each line of the list of steps should be added to your program as a comment.
3. When converting the algorithm to C++ program, first define variables for all the values used in the program. Use meaningful names for all the variables. If the value of the variable is known, initialize the variable.
4. Your program should follow the conventional programming style such as
 - a) align all statements properly
 - b) indent the statements within each pair of { }
 - c) insert blank lines to separate the variable definition statements and others
 - d) avoid using unnecessary parentheses, such as the ones (in red color) below:

```
avg = ( (num1+num2+num3) / 3.0 );
```

```
....
```

5. When you finish your program, print both the program (the text) and the running result of the program (the printout of the program).

Problem 1:

Write a program that will prompt the user to enter two floating point numbers and then read them (the first read into a variable called **first** and the second read into a variable called **second**). Your main function should print out the values of these two numbers immediately after they are read. Your main function should then call a function named **swap** with the actual parameters **first** and **second**. The **swap** function having formal parameters **number1** and **number2** should swap the value of the two variables. Your main function should print the values of **first** and **second** again after the **swap** function returns.

Note: Your program must contain all three parts of the function **swap**: the function prototype, the function call and the function definition.

Sample Run:

```
Enter the first number
Then hit enter
80
```

```
Enter the second number
Then hit enter
70
```

```
You input the numbers as 80 (first) and 70 (second).
After swapping, the first number has the value of 70
and the second number has the value of 80.
```

Problem 2:

Write a program that will convert miles to kilometers or kilometers to miles. The main function will prompt the user to input both a distance number and a choice of whether that number is in miles to be converted to kilometers or kilometers to be converted to miles. The main function will check for the validity of the input choice value. The main function **MUST** call a **value returning function** to convert the distance either to miles or kilometers. You must create a function to do the conversion. **You will create only ONE function to do both conversions.** The function should take two parameters: 1. Distance, 2. Choice. After the function call returns, the main function will then print out the converted result. You may use the following values for conversions.

1 kilometer = .621 miles 1 mile = 1.61 kilometers

Sample Run:

```
1. Convert miles to kilometers
2. Convert kilometers to miles
3. Quit
```

Your choice? **1**

Please input the miles to be converted: **120**

120 miles = 193.2 kilometers

```
1. Convert miles to kilometers
2. Convert kilometers to miles
3. Quit
```

Your choice? **2**

Please input the kilometers to be converted: **235**

235 kilometers = 145.935 miles

```
1. Convert miles to kilometers
2. Convert kilometers to miles
3. Quit
```

Your choice? **5**

Invalid input!

```
1. Convert miles to kilometers
2. Convert kilometers to miles
3. Quit
```

Your choice? **3**

The program is finishing. Thank you for using the program!