Computer Science 1081 – Assignment #06

Program #1

Download the “Random.txt” file (file name may differ). This file contains a long list of random numbers. Copy the file to your hard drive, place it in the same folder as the CPP file you are coding (inside the solution folder) and then write a program that opens the file, reads all the numbers from the file, and calculates the following:

A) The number of numbers in the file

B) The sum of all the numbers in the file (a running total)

C) The average of all the numbers in the file.

The program should display the number of numbers found in the file, the sum of the numbers, and the average of the numbers.

- You will need a text file called “Random.txt” to complete this program. You can download it from D2L. Save it in the same directory as your source file.
- Make sure you test for a successful file open operation, if it fails to open, display an error message – if you have trouble getting your program to open the file, please email me
- Use a constant to refer to the file location.

Sample Output (values may differ from actual results for this particular random file):

```
Number of numbers: 201
Sum of the numbers: 106042
Average of the numbers: 527.572
Press any key to continue . . .
```

Program #2

Write a program that asks the user to enter an item’s wholesale cost and its markup percentage. It should then display the item’s retail price. For example: if an item’s wholesale cost is 5.00 and its markup percentage is 100%, then the item’s retail price is 10.00. If the item’s wholesale cost is 5.00 and its markup percentage is 50% , then the item’s retail price is 7.50.

The program should have a function named calculateRetail that receives the wholesale cost and markup percentage as arguments and returns the retail price of the item. Do not accept negative values for either the wholesale cost of the item or the markup percentage.

- The calculateRetail function should not print anything to cout, instead it returns the value and the main-function does the printing

Sample Output:

```
Enter the item's wholesale cost: 14.25
Enter the item's markup percentage: 95
The item's retail price is $27.79
Press any key to continue . . .
```

Program #3

When an object is falling because of gravity, the following formula can be used to determine the distance the object falls in a specific time period: d = (1/2)gt^2. The variables in the formula are as follows: d is the distance in meters, g is 9.8, and t is the amount of time, in seconds, that the object has been falling.

Write a function named fallingDistance that accepts an object's falling time (in seconds) as an argument. The function should return the distance, in meters, that the object has fallen during that time interval. Write a program that demonstrates the function by calling it in a loop that passes the values 1 through 10 as arguments and displays the return value.

- Display two significant digits for the distance fallen (i.e. 17.00, not 17)
- Do not print data from the fallingDistance function, only return a value.

Sample Output:

```
Seconds   Distance
===============================
 1          4.90 meters
 2         19.60 meters
 3         44.10 meters
 4         78.40 meters
 5        122.50 meters
 6        176.40 meters
 7        240.10 meters
 8        313.60 meters
 9        396.90 meters
10        490.00 meters
Press any key to continue . . .
```

Program #4

Write a function named coinToss that simulates the tossing of a coin. When you call the function, it should generate a random number in the range of 1 through 2. If the random number is 1, the function should display “heads”. If the random number is 2, the function should display “tails”. Demonstrate the function in a program that asks the user how many times the coin should be tossed and then simulates the tossing of the coin that number of times.

- The coinToss function is a non-returning function, thus you should print the result of the coin toss from the function itself
- Use the C++ Random Number Generator
  - Generates a random number between 0 and 32767.
  - You have to seed the generator before using it. A good seed value is the current time. Use the following code to seed your generator.

```
unsigned seed = time(0);
srand(seed);
```

  - To generate a random number, between 1 and 2, use the expression

```
number = 1 + rand() % 2;
```

  - You will need to include the following header files:

```
#include <cstdlib>
#include <ctime>
```

Sample Output:

```
How many tosses should I make? 7
 1:     tails
 2:     heads
 3:     heads
 4:     heads
 5:     tails
 6:     heads
 7:     heads
Press any key to continue . . .
```