Reference: *Tony Gaddis Standard Version of Starting Out With C++ from Control Structures through Objects (9th Edition) Pearson, 2018 ISBN-13: 978-0134498379 ISBN-10: 0134498372*

Assignments – Chapter 5

16. Savings Account Balance

Write a program that calculates the balance of a savings account at the end of a period of time. It should ask the user for the annual interest rate, the starting balance, and the number of months that have passed since the account was established. A loop should then iterate once for every month, performing the following:

A) Ask the user for the amount deposited into the account during the month. (Do not accept negative numbers.) This amount should be added to the balance.

B) Ask the user for the amount withdrawn from the account during the month. (Do not accept negative numbers.) This amount should be subtracted from the balance.

C) Calculate the monthly interest. The monthly interest rate is the annual interest rate divided by twelve. Multiply the monthly interest rate by the balance, and add the result to the balance.

After the last iteration, the program should display the ending balance, the total amount of deposits, the total amount of withdrawals, and the total interest earned.

NOTE: If a negative balance is calculated at any point, a message should be displayed indicating the account has been closed and the loop should terminate.

18. Population Bar Chart

Write a program that produces a bar chart showing the population growth of Prairieville, a small town in the Midwest, at 20-year intervals during the past 100 years. The program should read in the population figures (rounded to the nearest 1,000 people) for 1900, 1920, 1940, 1960, 1980, and 2000 from a file. For each year it should

display the date and a bar consisting of one asterisk for each 1,000 people. The data can be found in the People.txt file.

Here is an example of how the chart might begin:

```
PRAIRIEVILLE POPULATION GROWTH
(each * represents 1,000 people)
1900 **
1920 ****
1940 *****
```

20. Random Number Guessing Game

Write a program that generates a random number and asks the user to guess what the number is. If the user's guess is higher than the random number, the program should display "Too high, try again." If the user's guess is lower than the random number, the program should display "Too low, try again." The program should use a loop that repeats until the user correctly guesses the random number.

21. Random Number Guessing Game Enhancement

Enhance the program that you wrote for Programming Challenge 20 so it keeps a count of the number of guesses that the user makes. When the user correctly guesses the random

number, the program should display the number of guesses.

24. Using Files—Numeric Processing

If you have downloaded a file named Random.txt in the Chapter 05 module. This file contains a long list of random numbers. Copy the file to your hard drive 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.

26. Using Files—Savings Account Balance Modification

Modify the Savings Account Balance program described in Programming Challenge 16 so that it writes the final report to a file.