Ekaba Bisong
Programming in C++
University of Calabar

Take-Home Exercise #1
June 03, 2015

Adapted from C++ How To Program edited for our own purposes

Take-Home Exercise

Factorial
The factorial of a nonnegative integer n is written n! (pronounced "n factorial") and is defined as follows:

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 1 \quad \text{(for values of } n \text{ greater than 1)}$$

and

$$n! = 1 \quad \text{(for } n = 0 \text{ or } n = 1\text{)}.$$

For example, 5! = 5 ∗ 4 ∗ 3 ∗ 2 ∗ 1, which is 120. Use while statements in each of the following:

a) Write a program that reads a nonnegative integer and computes and prints its factorial.
b) Write a program that estimates the value of the mathematical constant ***e*** by using the formula:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \ldots$$

Prompt the user for the desired accuracy of ***e*** (i.e., the number of terms in the summation).
c) Write a program that computes the value of e^x by using the formula

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$$

Prompt the user for the desired accuracy of e (i.e., the number of terms in the summation).

Challenging: (Enforcing Privacy with Cryptography)
The explosive growth of Internet communications and data storage on Internet-connected computers has greatly increased privacy concerns. The field of cryptography is concerned with coding data to make it difficult (and hopefully—with the most advanced schemes—impossible) for unauthorized users to read. In this exercise you'll investigate a simple scheme for encrypting and decrypting data. A company that wants to send data over the Internet has asked you to write a program that will encrypt it so that it may be transmitted more securely. All the data is transmitted as four-digit integers. Your application should read a four-digit integer entered by the user and encrypt it as follows: Replace each digit with the result of adding 7 to the digit and getting the remainder after dividing

the new value by 10. Then swap the first digit with the third, and swap the second digit with the fourth. Then print the encrypted integer. Write a separate application that inputs an encrypted four-digit integer and decrypts it (by reversing the encryption scheme) to form the original number. [Optional reading project: Research "public key cryptography" in general and the PGP (Pretty Good Privacy) specific public key scheme. You may also want to investigate the RSA scheme, which is widely used in industrial-strength applications.]