CSE 20212 Fundamentals of Computing II
Spring 2008

Lab handout for Week of February 11

Objectives

1. Learn more about container classes and STL strings
2. A smidgen about const objects and friend functions
3. Utilize operator overloading in varied settings
4. Incorporate code into your programs based on an interface file
5. Have fun!

Pre-lab assignment

1. (5 points) Problem 10.10 in D & D.

2. (5 points) Problem 11.13 in D & D. Also submit a driver program that tests all features of the class you developed with minimal input from the user if possible.

3. (5 points) Problem 11.15 in D & D. You are encouraged to use your from your the previous assignment for this problem. As with 11.13, please submit a driver program that tests all features requesting minimal input.

4. (5 points) Develop a simple class called "Codon" that stores a string of length 3 from the alphabet {A, C, G, T}. Provide a default constructor that initializes the private data member string to "TGA" and a non-default constructor for this class, which accepts a string as an argument. Incorporate data validation: any string whose length is not equal to 3 or whose characters are not from the alphabet {A, C, G, T} will result in the empty codon "---" being stored in the object. Overload the input and output operators to work directly on Codon objects. Further, overload the == and != operators to work with this class. Please also provide traditional get/set functions for this class. Finally, provide a test program that asks the user for a single string, places it into a codon object, and displays the contents of the codon object using cout.

In-lab activities

1. (1 point) Report to lab **on time**. Attendance will be taken at the scheduled lab time.
2. Implement a new container class called "Gene" that contains a dynamically allocated array of "Codons". Your test program should read in a single line from a file containing codons, determine its length using the STL string function **length()**, and utilize a non-default constructor to allocate the memory needed. The default Gene constructor should allocate space for 1000 codons.
3. Overload the = operator such that you can load the STL string read from the file into a gene object directly.
4. Define two constant codon objects in your program. Start is defined as "ATG" and

stop is defined as "TGA"

5. Develop code that will return codons stored in a gene object one at a time via a member function called "nextCodon()". Accomplish this goal by storing the current codon index as a private data member in the class Codon. Accompany this new iterator with a "isEmpty()" member function that returns 1 if and only if the current codon index is equal to the total number of codons. Note: if nextCodon is called and no codons are left, nextCodon should return the empty codon "---";
6. For this class, an "open reading frame" is defined as the region between start ("ATG") and stop ("TGA") codons. Develop a test program that iterates through codons stored in a gene object until the gene is empty, and outputs the length of all open reading frames from a string stored in a file. This must be accomplished by using the overloaded "==" operator for the codon class and using the two constant codons Start and Stop. A sample file will be posted closer to the date of the lab.
7. Download the "ribosome" interface and implementation from the course website. Review the interface file and determine how you can use it with your gene/codon classes (hint: you may have to use a get function).
8. Incorporate the "translate" member function of the Ribosome class into your program, which accepts the string of length 3 from the user and returns a character corresponding to its amino acid character. Test this new function by displaying the protein sequences corresponding to the open reading frames discovered in #6.
9. (4 points) Flag down the lab TA and have them examine and check off your work.

Post-lab (due at the start of next week's lab)

Write and submit (in your dropbox) a lab report with the following sections.

- (5 points) Functional code for the Codon and Gene classes, as well as the test program that calculates the lengths of all open reading frames w/ amino acid sequences too.
- (5 points) In order to differentiate the sample genes used in this lab from real genes in nature, we have placed a "watermark" that is composed of additional codons into the class examples. Implement a "diff" function that returns a synthetic gene composed of all codons not found in the reference gene. Convert this new gene into the amino acid alphabet to uncover the secret CSE messages inside.
- (5 points) A one-page (or so) description of the problem, your solution, and the secret messages encoded in the class examples. If you were going to create a watermark, how would you go about doing it? High level ideas are sufficient here.

Code is expected to be well-commented, and the narrative portions of your report must be written professionally (*i.e.*, complete sentences, correct grammar and punctuation, consistent tense and voice, formal tone).