

2800ICT Assignment Milestone

This assignment requires you to solve problems and write software independently, and to integrate various aspects of C++ programming that you have learned this trimester. You must write ALL the C++ program code. This program should maximize the use of object-oriented C++ programming.

Problem Description

Top secret agent Claude L. Evers has a unique way to send encoded messages. The message is implanted in a square grid of letters as follows: starting at the middle letter at the left-most side of the grid he writes the message along the diagonal heading to the top of the grid. After he hits the top, he then continues the message on a diagonal heading right until he hits the right-hand side. He continues this way, tracing out a diamond until he reaches the left side again. At this point he continues the message in the diamond nested inside the diamond he just filled in, and after this the next diamond inside, and so on. Any leftover squares both inside and outside the original diamond are filled with random letters.

The grids below show the steps in encoding the message (ignoring the blank spaces):

GENERAL TSO NEEDS CHICKEN NOW

The first grid shows the encoding after the first diamond has been filled in. The second, third and fourth grids show successively smaller diamonds being filled in. Finally, the last grid shows the final message after random letters have been placed in the empty squares around the grid.

(1) (2) (3) (4)

Note that if the original message had not filled in the entire set of diamonds, then random letters would have been used to fill those empty slots as well (see Sample 2 for an example of this when encoding GENERAL TSO HATES CHICKEN). To send the message, Evers concatenates all the rows together, top-to-bottom. The above grid would be sent as

THEESEARENSRNTAEDNCACGEEWNHLTEKOITUALNCSLYRANODOM

Now, your task is to develop a program capable of both encoding and decoding any message.

The encoded message consists of a single string of uppercase alphabetic letters. The length (<1000) of the encoded message will be a square of an odd integer. Note that the decoded message might contain some random letters at the end if the original message did not fill the entire set of diamonds.

Some encoded and its decoded messages are like below:

Sample 1,

- Encoded: THEESEARENSRNTAEDNCACGEEWNHLTEKOITUALNCSLYRANODOM
- Decoded: GENERALTSONEEDSCHICKENNOW

Sample 2,

- Encoded: WHAETHERENSRALLEENCAYGTIELHLKAKEITSARHCSEBIGMOACS
- Decoded: GENERALTSOHATESCHICKENIEL

Sample 3,

- Encoded: JOMHNPARTAILUUHLASERINWGO
- Decoded: IAMTHEWALRUSU

Requirements:

Your task is to develop an object-oriented C++ program with a textual user interface like below. After running each option, the interface should display after its output.

```
*****
* 1, Enter a message                                     *
* 2, Load a message from a file                         *
* 3, Encode a message                                    *
* 4, Decode a message                                    *
* 5, Save the message & decoded message to a file.      *
* 6, Quit                                                *
*****
```

- This program can always continue, even if a user enters incorrect data, until the user quit it. This means that the system should be able to handle various exceptions that may occur during execution.
- For option 2 and 5, allow the user to enter a file name or use a default file name.
- For option 3, when developing the encoding scheme, provide two ways for setting the size of the square grid:
 - one is to allow the user to enter the grid size,
 - one is to let the program to decide a suitable size for the square grid. Note that there may be multiple suitable sizes for the square grid.
- For option 4, your task is to implement the decoding scheme. Your program should accurately decode the encoded messages provided in Sample 1 through Sample 3.
- Produce a UML for each class used in your program. Each UML must clearly identify:
 - The name of the class
 - The name and data type of each member variable
 - For each member function
 - Its name
 - Its return data type
 - For each parameter: its name and data type
 - All member variables and functions should clearly show their visibility (i.e. private(-)/public(+)/protected(#))
 - Inheritance relationship if applicable
- Be well written (sensible variable names, indentation, no too long functions, appropriate comments)

Submit:

1, **all C++ source code:** *.cpp and *.hpp if your code is organized into separate files.

2, **week11.txt:** a txt file contains all the source code.

3, **week11.docx or week11.pdf, this document should include:**

- UML for each class
- Screenshot for running each option in the textual user interface, including normal and abnormal inputs. For example, the normal and abnormal inputs are like below:

#option	Normal Input		Abnormal Input
1	A message		Empty input
2	File name		Wrong file name
3	A message	Grid size by user	Empty message input
		Grid size by program	Grid size by user but not a number
4	The three samples		The length of the encoded message is not a square of an odd integer.
5	File name		Empty input
6	Desired character		Not desired character

Please refer to the submission page for the Marking Rubric.