SCHOOL OF COMPUTING (SOC)

Diploma in Applied AI and Analytics

ST1507 DATA STRUCTURES AND ALGORITHMS (AI)

2024/2025 SEMESTER 1
ASSIGNMENT ONE (CA1)
~ Morse Code Message Analyzer ~

Objective of Assignment

To practice what you have learnt in the module on data structures, algorithms, and object-oriented programming by developing a Morse Code Message Analyzer.

Instructions and Guidelines:

1. This is an individual assignment, and it accounts for **30%** of your final grade.

2. The submission date is **Tuesday 4 June at 5:00 pm**.

3. The development will be carried out in Python using Anaconda. You should only make use of those libraries that already ship with Anaconda Jupyter. You should not make use of the collection library.

4. The interviews will be conducted during the DSAA lessons in week 11. You are expected to explain your code and program logic. Take note that the interview is compulsory.

5. **50% of marks** will be deducted for submission of assignment within **ONE** calendar day after the deadline. **No marks shall** be awarded for assignments submitted **more than one day** after the deadline.

Warning: Plagiarism means passing off as one's own the ideas, works, writings, etc., which belong to another person. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turning it in as your own, even if you would have the permission of that person.
Plagiarism is a serious offence, and if you are found to have committed, aided, and/or abetted the offence of plagiarism, disciplinary action will be taken against you. If you are guilty of plagiarism, you may fail all modules in the semester, or even be liable for expulsion.

Context of the application

While studying historical morse code message data scientists may resort to data analysis to categorize and classify different types of messages. One method would be to focus on only the keywords in the message and omit those more common stop words, such as to mention a few; *'a', 'about', 'above', 'across', 'after',…,'yet', 'you', 'your', 'yours', 'yourself', 'yourselves'.* Each message could then be used to construct a 'keyword frequencies signature' comprising the frequencies of the sorted keywords.

For instance, the following two morse code messages, in terms of keyword frequencies, would have similar keyword frequencies signatures. This could potentially serve as an input for a classification algorithm to group the two messages together in the same class.

Morse Message One:

Morse Code
-.-.,---,--,., ,.-,-, ,---,-.,-.-.,., ,.--,., ,....,.-,...-,., ,...,-,.-.,..-,-.-.,-.-, ,.-,-., ,..,-.-.,.,-...,.,.-.,--. .--,., ,.-,.-.,., ,...,..,-.,-.-,..,-.,--. .-.,.,.--.,.,.-,-, ,.--,., ,.-,.-.,., ,...,..,-.,-.-,..,-.,--.
Plain Text
COME AT ONCE WE HAVE STRUCK AN ICEBERG WE ARE SINKING REPEAT WE ARE SINKING
Sorted Keywords frequencies

SINKING(2)
COME(1)
ICEBERG(1)
REPEAT(1)
STRUCK(1)

Morse Message Two:

Morse Code
...,---,..., ,.--,., ,.-,.-.,., ,...,..,-.,-.-,..,-.,--. .--,., ,....,.-,...-,., ,...,-,.-.,..-,-.-.,-.-, ,.-,-., ,..,-.-.,.,-...,.,.-.,--. .--,., ,.-,.-.,., ,...,..,-.,-.-,..,-.,--., ,-...,-.--, ,-,....,., ,....,.,.-,-..
Plain Text
SOS WE ARE SINKING WE HAVE STRUCK AN ICEBERG WE ARE SINKING BY THE HEAD
Sorted Keywords frequencies

SINKING(2)
HEAD(1)
ICEBERG(1)
SOS(1)
STRUCK(1)

Basic functionality of the application

Your job is to implement a *Morse Code Message Analyzer* Application that serves the following four purposes:

1. Allows the user to convert plain text to morse code.
2. Allows the user to convert morse code to plain text.
3. Allows the user to generate a morse code frequency report.
4. Allows the user to generate a morse code keyword frequency graph.

An example of morse code encoding is provided next:

The plain text:
THIS IS A SOS

Will be converted into morse code:
-,....,..,..., ,..,..., ,.-, ,...,---,...

Take note:

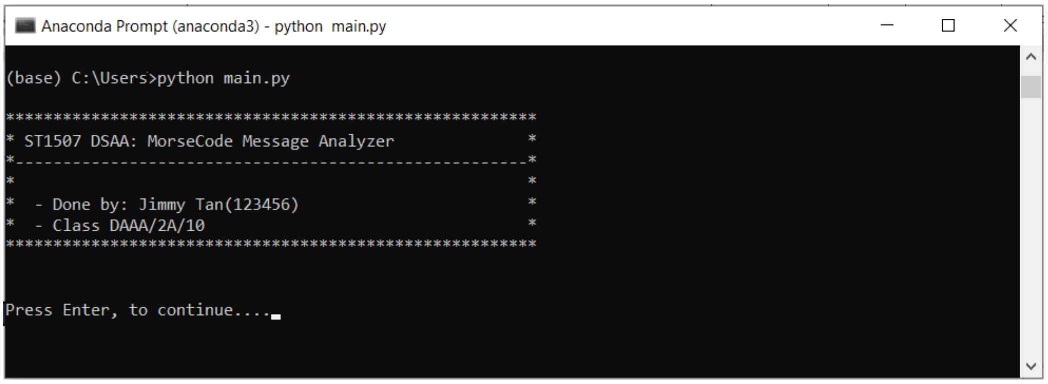
- A Morse Code lookup table is provided in appendix A
- For our conversion, morse words will be separated by spaces and morse letters will be separated by commas.

Starting the application

Your application must be able to start from the Anaconda Command Prompt by typing:

```
python main.py
```

It will then display a tittle bar with essential information, next it waits for the user to press the enter key before the application continues.

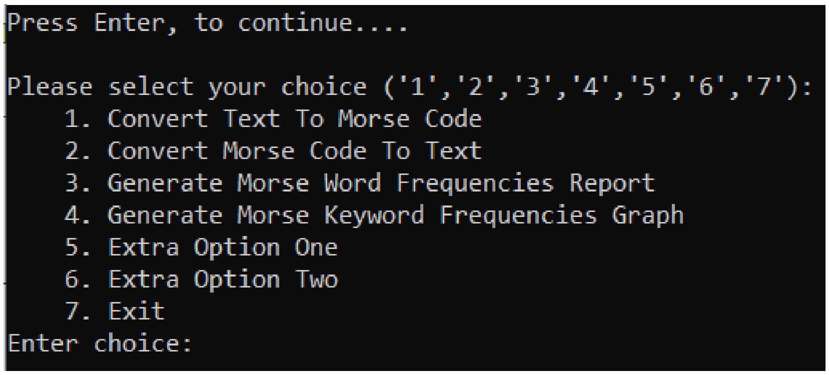


```
(base) C:\Users>python main.py

********************************************************
* ST1507 DSAA: MorseCode Message Analyzer              *
*------------------------------------------------------*
*                                                      *
*  - Done by: Jimmy Tan(123456)                        *
*  - Class DAAA/2A/10                                  *
********************************************************


Press Enter, to continue....
```

- You are required to follow the above format.
- Please, ensure that you display your name, student ID and class in the format as is shown in the example.

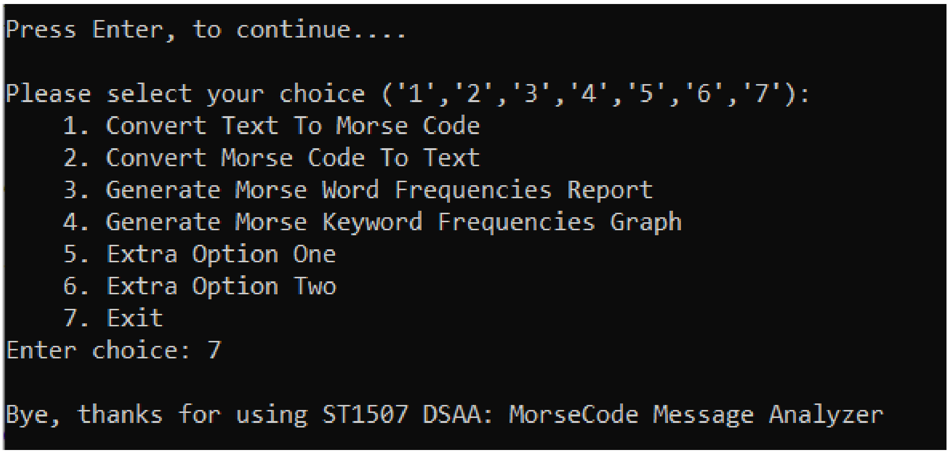
Selection menu

When the application continues, the user will be presented with a menu as is shown below. The menu allows the user to choose from 7 options.



```
Press Enter, to continue....

Please select your choice ('1','2','3','4','5','6','7'):
    1. Convert Text To Morse Code
    2. Convert Morse Code To Text
    3. Generate Morse Word Frequencies Report
    4. Generate Morse Keyword Frequencies Graph
    5. Extra Option One
    6. Extra Option Two
    7. Exit
Enter choice:
```

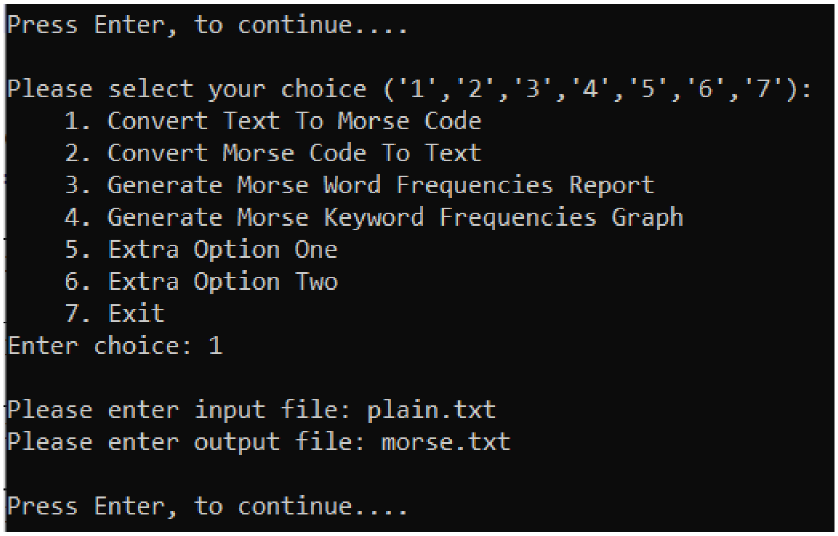
- You are required to follow the above format for the menu display.
- Option 5 and 6 are reserved as extra options that you will need to design and implement yourself.
- The user will be able to repeatedly select options from the menu.
- The application will terminate once the user selects option 7 Exit (displaying an Exit message as shown below).



```
Press Enter, to continue....

Please select your choice ('1','2','3','4','5','6','7'):
    1. Convert Text To Morse Code
    2. Convert Morse Code To Text
    3. Generate Morse Word Frequencies Report
    4. Generate Morse Keyword Frequencies Graph
    5. Extra Option One
    6. Extra Option Two
    7. Exit
Enter choice: 7

Bye, thanks for using ST1507 DSAA: MorseCode Message Analyzer
```

Option 1: Convert Text to Morse Code

The user may select option ‘1’ to read plain text from an input file that will then be converted into morse code. The user will be prompted to type in both an input and output file, as shown below. The resulting morse code will be written back to the output file.



```
Press Enter, to continue....

Please select your choice ('1','2','3','4','5','6','7'):
    1. Convert Text To Morse Code
    2. Convert Morse Code To Text
    3. Generate Morse Word Frequencies Report
    4. Generate Morse Keyword Frequencies Graph
    5. Extra Option One
    6. Extra Option Two
    7. Exit
Enter choice: 1

Please enter input file: plain.txt
Please enter output file: morse.txt

Press Enter, to continue....
```

Next is an example of how the application would convert plain text into morse code.

Input file: plain.txt

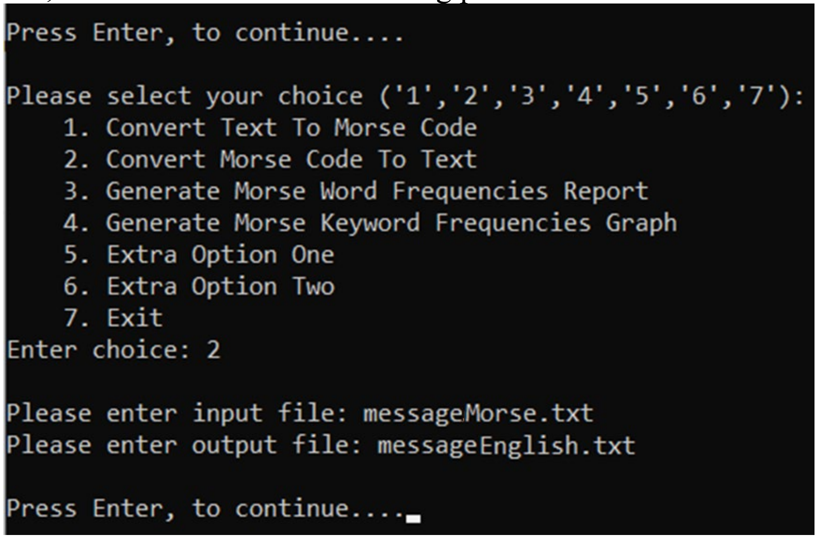
```
HELP US SOS SOS SOS
OUR SHIP HAS HIT AN ICEBERG
PLEASE HELP US
THIS IS A SOS
OUR SHIP IS SINKING
WE HIT AN ICEBERG
THIS IS AN SOS
```

Output file: morse.txt

```
....,.,.-..,.--., ,..-,..., ,...,---,..., ,...,---,..., ,...,---,...
---,..-,.-., ,...,....,..,.--., ,....,.-,..., ,....,..,-, ,.-,-., ,..,-.-.,.,-...,.,.-.,--.
.--.,.-..,.,.-,...,., ,....,.,.-..,.--., ,..-,...
-,....,..,..., ,..,..., ,.-, ,...,---,...
---,..-,.-., ,...,....,..,.--., ,..,..., ,...,..,-.,-.-,..,-.,--.
.--,., ,....,..,-, ,.-,-., ,..,-.-.,.,-...,.,.-.,--.
-,....,..,..., ,..,..., ,.-,-., ,...,---,...
```

Option 2: Morse Code to Text

The user may select option ‘2’ to read morse code from an input file that will then be converted into plain text. The user will be prompted to type in both an input and output file, as shown below. The resulting plain text will be written back to the output file.



```
Press Enter, to continue....

Please select your choice ('1','2','3','4','5','6','7'):
    1. Convert Text To Morse Code
    2. Convert Morse Code To Text
    3. Generate Morse Word Frequencies Report
    4. Generate Morse Keyword Frequencies Graph
    5. Extra Option One
    6. Extra Option Two
    7. Exit
Enter choice: 2

Please enter input file: messageMorse.txt
Please enter output file: messageEnglish.txt

Press Enter, to continue....
```

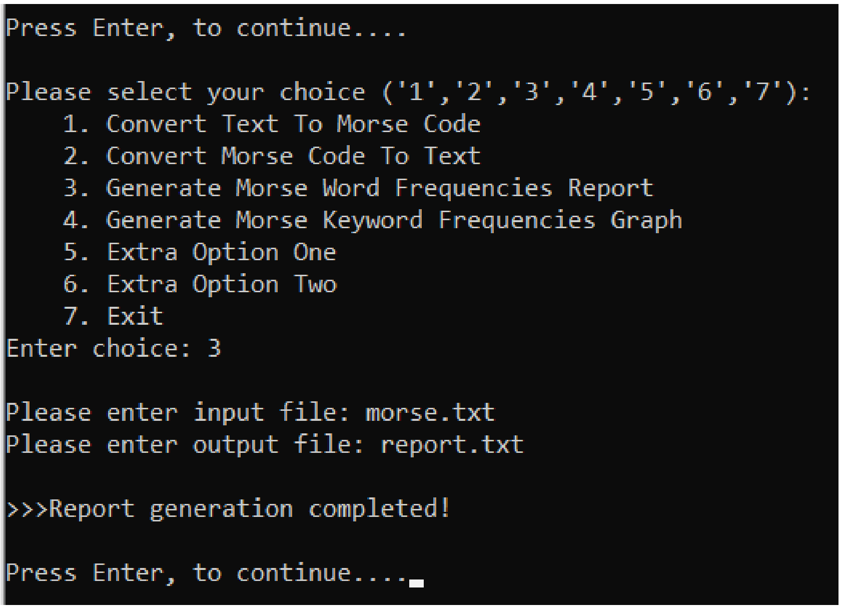
Next is an example of how the application would convert morse code into plain text.

Input file: messageMorse.txt
...,---,..., ,.--,., ,-.,.,.,-.., ,..-,.-.,--.,.,-.,-, ,--,.,-..,..,-.-.,.-,.-.., ,....,.,.-..,.--. ...,.,-.,-.., ,.-, ,--,.,-..,..,-.-.,.-,.-.., ,-,.,.-,--, ,-.,---,.-- ...,---,..., ,...,---,...

Output file: messageEnglish.txt
SOS WE NEED URGENT MEDICAL HELP SEND A MEDICAL TEAM NOW SOS SOS

Option 3: Generate Morse Word Frequency Report

The user may select option '3' to generate a Morse Word Frequencies Report. The user will be prompted to enter an input file (morse code) and an output file (report). The application will then analyze the frequencies of morse code words and summarize the findings in the report.



```
Press Enter, to continue....

Please select your choice ('1','2','3','4','5','6','7'):
    1. Convert Text To Morse Code
    2. Convert Morse Code To Text
    3. Generate Morse Word Frequencies Report
    4. Generate Morse Keyword Frequencies Graph
    5. Extra Option One
    6. Extra Option Two
    7. Exit
Enter choice: 3

Please enter input file: morse.txt
Please enter output file: report.txt

>>>Report generation completed!

Press Enter, to continue....
```

The Morse Word Frequencies Report consists out of the following sections:

- **Time stamp indicating the date and time that the report was generated:**

  Example:

```
**********************************************
   REPORT GENERATED ON: 21-03-2024 14:41
**********************************************
```

- **Decoded morse text:**

  Example:

```
*** Decoded Morse Text
HELP US SOS SOS SOS
OUR SHIP HAS HIT AN ICEBERG
PLEASE HELP US
THIS IS A SOS
OUR SHIP IS SINKING
WE HIT AN ICEBERG
THIS IS AN SOS
```

- **Summary of morse words grouped, and sorted by frequencies:**

  Example:

```
*** Morse Words with frequency=> 1
[...,..,-.,-.-,..,-.,--.]=> SINKING(*)
[.--.,.-..,.,.-,...,.]=> PLEASE
[....,.-,...]=> HAS
[.--,.]=> WE
[.-]=> A

*** Morse Words with frequency=> 2
[..,-.-.,.,-...,.,.-.,--.]=> ICEBERG(*)
[....,.,.-..,.--.]=> HELP(*)
[...,....,..,.--.]=> SHIP(*)
[-,....,..,...]=> THIS
[---,..-,.-.]=> OUR
[....,..,-]=> HIT(*)
[..-,...]=> US

*** Morse Words with frequency=> 3
[..,...]=> IS
[.-,-.]=> AN

*** Morse Words with frequency=> 5
[...,---,...]=> SOS(*)
```

  Take note:
  - Morse words are sorted by frequency in ascending order.
  - Morse words with the same frequency will be sorted by length in descending order.
  - Morse words with the same frequency and same length will be sorted by alphabetically order of the plain text word.
  - Those morse words that are keywords are labeled with a postfix '(*)'
    (**Take note:** keywords are all those words that are not being identified as stop-words. You must make use of the list of stop-words that is provide on Brightspace in the file *stopwords.txt*)

- **List of keywords sorted by frequencies:**

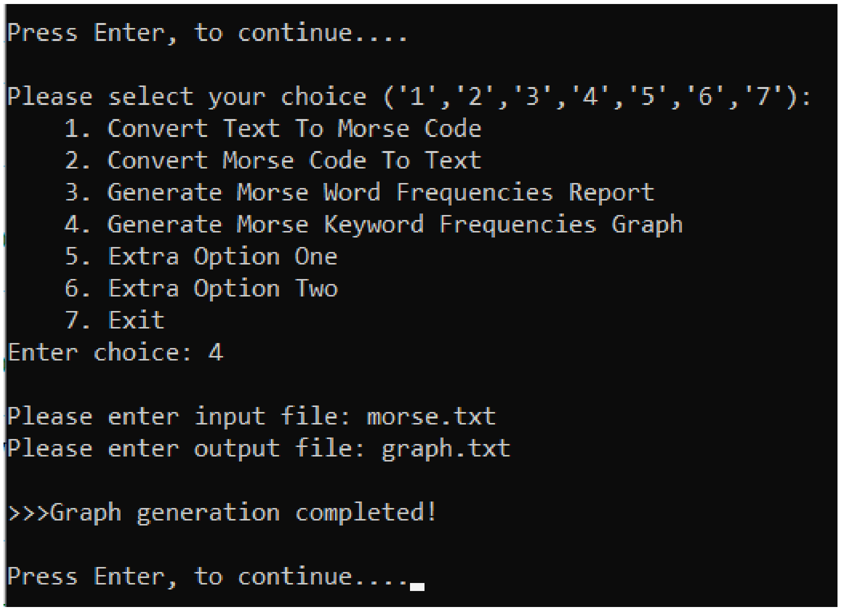
  Example:

```
*** Keywords sorted by frequency
SOS(5)
HELP(2)
HIT(2)
ICEBERG(2)
SHIP(2)
SINKING(1)
```

  - **Take Note**: Keywords are sorted by their frequencies in descending order. For keywords with the same frequencies, they will be sorted alphabetically.

Option 4: Generate Morse Keyword Frequency Graph

The user may select option '4' to generate a Morse Keyword Frequency Graph. The user will be prompted to enter an input file (morse code) and an output file (graph). The application will then analyze the frequencies of morse keywords and summarize the findings in the graph.



```
Press Enter, to continue....

Please select your choice ('1','2','3','4','5','6','7'):
    1. Convert Text To Morse Code
    2. Convert Morse Code To Text
    3. Generate Morse Word Frequencies Report
    4. Generate Morse Keyword Frequencies Graph
    5. Extra Option One
    6. Extra Option Two
    7. Exit
Enter choice: 4

Please enter input file: morse.txt
Please enter output file: graph.txt

>>>Graph generation completed!

Press Enter, to continue....
```

The Morse Keyword Frequencies Graph consists out of the following sections:

- **Time stamp indicating the date and time that the graph was generated:**

  Example:

```
**********************************************
   REPORT GENERATED ON: 21-03-2024 14:41
**********************************************
```

- **Bar graph depicting the frequencies of morse keywords:**

```
 *
 *
 *
 *         *         *         *         *
 *         *         *         *         *         *
------------------------------------------------------------
 .S        .H        .H        .I        .S        .S
 .O        .E        .I        .C        .H        .I
 .S        .L        .T        ,E        .I        .N
 ,         .P        .         -B        ,P        ,K
 -         ,         ,         .E        .         .I
 -         .         .         -R        .         .N
 -         ,         .         .G        .         ,G
 ,         .         ,         ,         .         -
 .         -         -         .         ,         .
 .         .                   ,         .         ,
 .         .                   -         .         -
           ,                   .         ,         .
           .                   .         .         -
           -                   .         -         ,
           -                   ,         -         .
           .                   .         .         .
                               ,                   ,
                               .                   -
                               -                   .
                               .                   ,
                               ,                   -
                               -                   -
                               -                   .
                               .
```

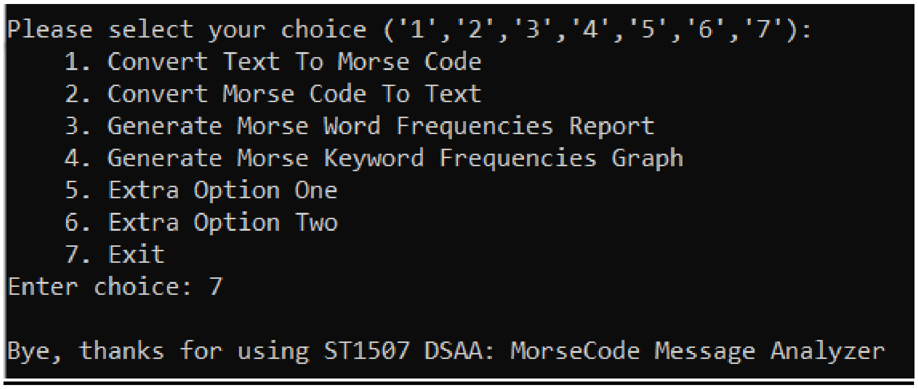
Take note:

- Keywords are listed from left to right, sorted by frequencies in descending order.
- Keywords with the same frequencies are sorted alphabetically.
- The bar height indicates the frequency of a keyword (for instance SOS appears 5 times).
- Each bar is labeled with the keyword in both morse code and plain text (both printed vertically).

(*) For an example of a complete morse word frequencies report and morse keyword frequency graph see Appendix B & C.   You are required to follow the format of thee examples.

Option 7: Exit

The user can repeatedly select Options 1 till 6. Option 7 is meant to exit the program. When the user Exits the application, you should show the message as is shown below.



```
Please select your choice ('1','2','3','4','5','6','7'):
    1. Convert Text To Morse Code
    2. Convert Morse Code To Text
    3. Generate Morse Word Frequencies Report
    4. Generate Morse Keyword Frequencies Graph
    5. Extra Option One
    6. Extra Option Two
    7. Exit
Enter choice: 7

Bye, thanks for using ST1507 DSAA: MorseCode Message Analyzer
```

Basic requirements:

- You are required to design and write the Python application using an object-oriented programing approach (OOP). You should leverage on your knowledge of encapsulation, function/operator overloading, polymorphism, inheritance etc.

- You may make use of Python's already built in data structures, such as list, tuple, dictionary etc., however you should refrain from using the classes from the collection library. Instead, you are required to write you own classes to support the various data structures that you may need (for instance for sorting you may need a class *SortedList*). Of course, you may refer to the lecture slides and lab tasks and expand further on those classes that we had previously developed and discussed in tutorials and lab sessions.

- To run the application there should be no need to install additional libraries, other than those that ship already with Anaconda.

- Your application should not have to rely on any connection to the Internet.

- The OOP classes that you develop must all be placed in separate python files.

- Pay attention to user input validation. Your application should not crash if a user types in the wrong input. Instead, when a user enters wrong input, you should notify the user, and allow him/her to enter again.

Deliverables

Your deliverables must include:

(a) Source Code

- You must submit all the python files (.py files) that make up your application. Ensure the code is complete, and that it can run directly from the Anaconda Prompt as:

```
python main.py.
```

(b) Report

A report (as pdf file) with a maximum **of 8** pages. This would exclude the cover page and the appendix with source listing and references. The report should contain:

a) Cover page with title and displaying your name, student ID and class.
b) Description, and user guidelines, on how to operate your application (please include screen shots of your application in action).
c) Description of how you have made use of the Object-Oriented Programming (OOP) approach. You may elaborate on the classes that you have developed, and discuss issues such as encapsulation, function/operator overloading, polymorphism, inheritance etc. Do include a class diagram that displays the relation between the various classes that you have developed.
d) Discussion on the data structures and algorithms that you have developed for your application. You may discuss issues such as, the performance of the algorithms in terms of Big (O). Do explain why you did develop certain data structures and explain why you deem these data structures suitable for the task(s) at hand. Do include a table summarizing all the data structures that you have been using (those that you have developed and those already built in Python).
e) Include a summary of the challenges that you have faced while developing the application. Provide a summary of the key takeaways and learning achievements that you have obtained from this project.
f) **All** your python **source code** must be included as an appendix at the end of your report. You may include in the appendix those references from literature or internet that you may have consulted.

(c) Academic Integrity Declaration for,

- Ensure that the form is duly filled in and signed.

Submission instructions

- Submit all the deliverables (Source Code, Report, Academic Integrity Declaration form) in the designated Brightspace drop box before the submission deadline.

- You must submit it as one Zipped folder (RAR will not be accepted, only zip).

  Label your submission as:

  CA1_Name_StudentID_Class.zip

  For example: *CA1_JimmyTan_12345_2A10.zip*

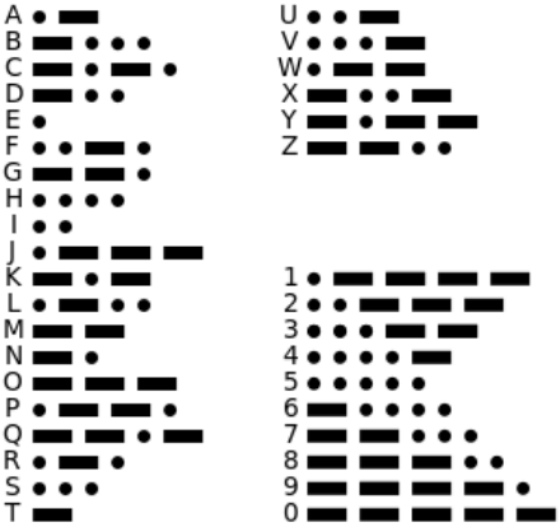
Assessment Criteria

The assignment will be assessed based on the following criteria:

Assessment criteria	Marks awarded
User Interface - GUI looks and operates as prescribed. - GUI supports appropriate user input validation and error handling.	Max 10
Basic functionality of the application (Option 1,2,3,4): Option 1/2: Morse Code Conversion: - Able to convert plain text into morse code correctly. - Able to convert morse code into plain text correctly. Option 3: Morse Code Frequency Report: - Able to generate the report will all the required sections displaying the correct analytical results. - The report is in the format as prescribed. Option 4: Morse keyword Frequency Graph: - Able to generate the graph with all the required sections displaying the correct analytical results. - The graph is in the format as prescribed.	Max 30
Two Extra Features (Option 5 & 6): - Features will be judged according to technical sophistication and usability of features.	Max 20
Programming techniques efficiency robustness, and readability of code: - Appropriate usage of classes and OOP technology. - Appropriate usage of data structures and algorithms. - Code is properly commented and neatly structured. - Application is free of crashes.	Max 20
Report: - The report follows the prescribed format. - The report is well written and comprehensive.	Max 10
Interview/Demonstration: - Ability to demonstrate and explain the application and the code clearly. - Q&A **(*) Multiplier on total score may be applied if student demonstrates poor understanding of code.**	Max 10
Grand Total	**100**

APPENDIX A- INTERNATIONAL MORSE CODE*

[*] By Rhey T. Snodgrass & Victor F. Camp, 1922, International Morse Code



A	● ▬	U	● ● ▬
B	▬ ● ● ●	V	● ● ● ▬
C	▬ ● ▬ ●	W	● ▬ ▬
D	▬ ● ●	X	▬ ● ● ▬
E	●	Y	▬ ● ▬ ▬
F	● ● ▬ ●	Z	▬ ▬ ● ●
G	▬ ▬ ●		
H	● ● ● ●		
I	● ●		
J	● ▬ ▬ ▬		
K	▬ ● ▬	1	● ▬ ▬ ▬ ▬
L	● ▬ ● ●	2	● ● ▬ ▬ ▬
M	▬ ▬	3	● ● ● ▬ ▬
N	▬ ●	4	● ● ● ● ▬
O	▬ ▬ ▬	5	● ● ● ● ●
P	● ▬ ▬ ●	6	▬ ● ● ● ●
Q	▬ ▬ ● ▬	7	▬ ▬ ● ● ●
R	● ▬ ●	8	▬ ▬ ▬ ● ●
S	● ● ●	9	▬ ▬ ▬ ▬ ●
T	▬	0	▬ ▬ ▬ ▬ ▬

APPENDIX B – Example of a Morse Word Frequency Report

```
**********************************************
   REPORT GENERATED ON: 21-03-2024 14:41
**********************************************

*** Decoded Morse Text
HELP US SOS SOS SOS
OUR SHIP HAS HIT AN ICEBERG
PLEASE HELP US
THIS IS A SOS
OUR SHIP IS SINKING
WE HIT AN ICEBERG
THIS IS AN SOS

*** Morse Words with frequency=> 1
[...,..,-.,-.-,..,-.,--.]=> SINKING(*)
[.--.,.-..,.,.-,...,.]=> PLEASE
[....,.-,...]=> HAS
[.--,.]=> WE
[.-]=> A

*** Morse Words with frequency=> 2
[..,-.-.,.,-...,.,.-.,--.]=> ICEBERG(*)
[....,.,.-..,.--.]=> HELP(*)
[...,....,..,.--.]=> SHIP(*)
[-,....,..,...]=> THIS
[---,..-,.-.]=> OUR
[....,..,-]=> HIT(*)
[..-,...]=> US

*** Morse Words with frequency=> 3
[..,...]=> IS
[.-,-.]=> AN

*** Morse Words with frequency=> 5
[...,---,...]=> SOS(*)

*** Keywords sorted by frequency
SOS(5)
HELP(2)
HIT(2)
ICEBERG(2)
SHIP(2)
SINKING(1)
```

APPENDIX C – Example of a Morse Keyword Frequencies Graph

```
************************************************************
         GRAPH GENERATED ON: 25-03-2024 09:40
************************************************************

 *
 *
 *
 *         *         *         *         *
 *         *         *         *         *         *
------------------------------------------------------------
 .S        .H        .H        .I        .S        .S
 .O        .E        .I        .C        .H        .I
 .S        .L        .T        ,E        .I        .N
 ,         .P        .         -B        ,P        ,K
 -         ,         ,         .E        .         .I
 -         .         .         -R        .         .N
 -         ,         .         .G        .         ,G
 ,         .         ,         ,         .         -
 .         -         -         .         ,         .
 .         .                   ,         .         ,
 .         .                   -         .         -
           ,                   .         ,         .
           .                   .         .         -
           -                   .         -         ,
           -                   ,         -         .
           .                   .         .         .
                               ,                   ,
                               .                   -
                               -                   .
                               .                   ,
                               ,                   -
                               -                   -
                               -                   .
                               .
```

-- End --