



FIT9133 Assignment #1

Building a Simple Morse Code Decoder with Python Programming

Semester 1 2018

Dr Jojo Wong
Lecturer, Faculty of IT.
Email: Jojo.Wong@monash.edu
© 2018, **Monash University**

March 8, 2018

Revision Status

\$Id: FIT9133-Assignment-01.tex, Version 1.0 2018/02/23 17:30 Jojo \$

\$Id: FIT9133-Assignment-01.tex, Version 1.1 2018/03/04 17:15 Gavin \$

\$Id: FIT9133-Assignment-01.tex, Version 1.2 2018/03/04 18:15 Jojo \$

\$Id: FIT9133-Assignment-01.tex, Version 1.3 2018/03/08 17:30 Jojo \$

Contents

1	Introduction	4
2	Morse Code Interpreter	5
2.1	Task 1: Building a Dictionary of Morse Code	6
2.2	Task 2: Reading Morse Code Sequences	7
2.3	Task 3: Decoding Morse Code Sequences	7
2.4	Task 4: Analysing Decoded Characters	7
3	Important Notes	9
3.1	Documentation	9
3.2	Marking Criteria	9
4	Submission	10
4.1	Deliverables	10
4.2	Academic Integrity: Plagiarism and Collusion	10

1 Introduction

This assignment is due on **29 March 2018 (Thursday) by 5:00pm**. It is worth **10% of the total unit mark**. A penalty of 5% per day will apply for late submission. Refer to the FIT9133 Unit Guide for the policy on extensions or special considerations.

Note that this is **an individual assignment** and **must be your own work**. Please pay attention to Section 4.2 of this document on the university policies for the *Academic Integrity, Plagiarism and Collusion*.

This first assignment consists of **four tasks** and all the tasks should be submitted as **separate Python programs** with supporting documentation on its usage. All the program files and any supporting documents should be compressed into one single file for submission. (The submission details are given in Section 4.)

Assessment: The first task (Task 1) is weighed 10% and the subsequent three tasks (Task 2 to Task 4) carry equal weightage of marks with 30% each.

2 Morse Code Interpreter

What is Morse Code?

Morse Code is a method of transmitting natural language text originally developed for telegraph communication. It was invented back in the 19 century by the inventor of telegraph *Sameul F. B. Morse*. The current International Morse Code system encodes a range of characters, including the ISO basic Latin alphabets (A–Z), some additional Latin letter representing accents, the Arabic numerals (0–9), and a small set of punctuations and procedural signals (known as prosigns).

Each character is presented in Morse Code as a unique sequence of ‘dot’ (.) and ‘dashes’ (_). See Figure 1 for the subset of characters represented in 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	— — — — —

Figure 1: International Morse Code representation (letters and numerals)
 [Source: https://en.wikipedia.org/wiki/Morse_code]

Each character is separated by a single space, and each word is separated by three spaces. Note that in transmission a space is represents a pause.

1	Word: WHO
2	Morse Sequence: . _ _ . . . _ _ _
3	
4	Sentence: WHO AM I
5	Morse Sequence: . _ _ . . . _ _ _ . _ _ . .

How Morse Code is represented in this assignment?

In this assignment, we are going to define a different set of representation for the Morse Code by using the binary digits. The 'dots' are represented by the digit 0 and the 'dashes' are represented by the digit 1. As for the spaces, they are represented by the character *. Note that our representation of Morse Code encodes only the standard 26 letters (i.e. 'A' to 'Z') and the 10 numerals (i.e. '0' to '9').

Table 1 defines the set of Morse Code representation used in this assignment.

Character	Morse Code	Character	Morse Code	Character	Morse Code
A	01	M	11	Y	1011
B	1000	N	10	Z	1100
C	1010	O	111	0	11111
D	100	P	0110	1	01111
E	0	Q	1101	2	00111
F	0010	R	010	3	00011
G	110	S	000	4	00001
H	0000	T	1	5	00000
I	00	U	001	6	10000
J	0111	V	0001	7	11000
K	101	W	011	8	11100
L	0100	X	1001	9	11110

Table 1: Morse Code representation for this assignment (letters and numerals)

2.1 Task 1: Buidling a Dictionary of Morse Code

In the first task, you are required to define or create a *dictionary* structure to represent each of the Morse Code characters (presented in Table 1) as a string sequence of binary digits (0 and 1). This structure will be used in the subsequent tasks of this assignment to decode any Morse Code sequences.

Display this Morse Code representation structure on the console (i.e. the terminal).

You should name the Python file of this task as Task1.py.

2.2 Task 2: Reading Morse Code Sequences

Extending from the first task, you are required to implement a sub-program that prompts the user to input any random sequences of Morse Code. The Morse Code sequences can be of any length but with the minimum length of 1.

Display all the Morse Code sequences entered by the user on the console (i.e. the terminal).

Note: Do not worry if the Morse Code sequences provided by the user do not represent any proper words or sentences in English. For the purpose of this assignment, we only focus on decoding individual characters but not words or sentences. Also note that the user should be allowed to input multiple Morse Code sequences not until the user indicates that he/she would like to terminate.

You should name the Python file of this task as `Task2.py`.

2.3 Task 3: Decoding Morse Code Sequences

Extending from Task 2, your task here is to implement the decoding sub-program. For each of the Morse Code sequences provided by the user, decode or translate the sequence into the corresponding characters, which can either be a letter ('A' to 'Z') or a digit ('0' to '9'). To perform the decoding process, you will have to make use of the dictionary for the Morse Code representation that you constructed in Task 1.

You should print each of the decoded sequences on the console (i.e. the terminal). If you encountered an invalid character (i.e. you were unable to decode a subsequence of the Morse Code to any of the 26 letters or the 10 numerals), display an error message indicating which subsequence is invalid.

Note: In terms of implementation, you may choose to perform the decoding after each Morse Code sequence has been read in. Alternatively, the decoding process can be performed after the user has entered all the Morse Code sequences.

You should name the Python file of this task as `Task3.py`.

2.4 Task 4: Analysing Decoded Characters

In the final task, you are going to analyse the decoded characters. Again, you should implement this task by extending Task 3 (i.e. you should have completed the previous three tasks).

before attempting this final part of the assignment.)

The task here is to find out the number of occurrences for each of the characters (a letter or a numeral) from the decoded sequences. For each of the Morse Code sequences decoded, present each of the 26 letters and the 10 numerals along with the number of occurrences on the console. You should then present the total number of occurrences for each character ('A' to 'Z' and '0' to '9') based on all the sequences entered by the user.

You should name the Python file of this task as `Task4.py`.

3 Important Notes

3.1 Documentation

Commenting your code is essential as part of the assessment criteria (refer to Section 3.2). You should also include comments at the beginning of your program file, which specify your name, your Student ID, the start date and the last modified date of the program, as well as with a high-level description of the program. In-line comments within the program are also part of the required documentation.

3.2 Marking Criteria

The assessment of this assignment will be based on the following marking criteria. The same marking criteria will be applied on both tasks:

- 60% for working program — functionality;
- 10% for code architecture — algorithms, data types, control structures, and use of libraries;
- 10% for coding style — clear logic, clarity in variable names, and readability;
- 20% for documentation — program comments and user documentation.

Note: As part of the assessment, you are required to attend an interview conducted by your tutor in the prac class in the following week after the assignment has been submitted. (That means if you missed the interview, your assignment will not be graded.)

4 Submission

There will be NO hard copy submission required for this assignment. You are required to submit your assignment as a .zip file named with your Student ID. For example, if your Student ID is 12345678, you would submit a zipped file named 12345678_A1.zip. Note that marks will be deducted if this requirement is not strictly complied with.

Your submission is via the assignment submission link on the FIT9133 S1 2018 Moodle site by the deadline specified in Section 1, i.e. **29 March 2018 (Thursday) by 5:00pm**.

4.1 Deliverables

Your submission should contain the following documents:

- A completed the assignment cover sheet for online submission available on the FIT9133 S1 2018 Moodle site.
- An user documentation (of 2 pages) in PDF format with clear and complete instructions on how to run your programs. (Note that your programs must at least run on the computers in the University's computer labs. Any submission that does not run accordingly will receive no marks.)
- Electronic copies of ALL your files that are needed to run your programs.

Marks will deducted for any of these requirements that are not strictly complied with.

4.2 Academic Integrity: Plagiarism and Collusion

Plagiarism means to take and use another person's ideas and or manner of expressing them and to pass them off as your own by failing to give appropriate acknowledgement. This includes materials sourced from the Internet, staff, other students, and from published and unpublished works.

Collusion means unauthorised collaboration on assessable work (written, oral, or practical) with other people. This occurs when you present group work as your own or as the work of another person. Collusion may be with another Monash student or with people or students external to the University. This applies to work assessed by Monash or another university.

It is your responsibility to make yourself familiar with the University's policies and procedures in the event of suspected breaches of academic integrity. (Note: Students will be asked to attend an interview should such a situation is detected.)

The University's policies are available at: <http://www.monash.edu/students/academic/policies/academic-integrity>