

The Leeds Challenge – Morse Code

We present a challenge that poses a "real world" problem. Listening to radio transmissions and detecting and decoding communications is a tricky problem that computers are being used to tackle: just ask the folks at SETI! Decoding sound waves is a bit tricky for a one hour assignment, so we've simplified the problem to use plain text. And to make it really easy we've assumed a noise-free communication channel.

Our (Python) solution, including the bonus marks task is about 80 lines long. Our Java is a bit longer, and we recommend looking at some features of Java 5 that make the whole thing rather easier to complete.

Here we go ...

Morse Code is a simple system for sending messages, usually over radio. Letters and other symbols are encoded as sequences of "dots" and dashes". These can be transmitted either as sounds or flashes of light. A short gap in transmission indicates the end of a sequence representing a letter and a longer gap indicates the end of a word.

The file `morse.txt` on the CD contains the encoding represented as ASCII text. The left column is the character, and the right column is the corresponding encoding. No other characters are defined in Morse Code. Notice that Morse Code makes no distinction between the case of letters, which is why the provided file includes only lower case.

Write one or more classes that can translate messages into Morse code and back again.

A message is a single paragraph of plain text, with no blank lines, which should be provided to your code as a single string object. The encoded form of the message need not be a string, but it must be printable and it must preserve line endings in the original message, such that decoding of the encoded form yields the original message (ignoring letter case).

Provide, as part of your solution, a small test program that:

- Reads a message from a text file named on the command line.
- Encodes that message and prints the result.
- Decodes the encoded message and prints the result.
- Demonstrates that the decoded message is equal to the original (ignoring letter case).

Marks will be awarded for:

- Correctness.
- Elegance of design.
- Efficient use of programming language features.
- Coding style.

Significant bonus points will be awarded to solutions that can render encoded messages as text strings in which:

- Each encoded letter is separated from the next by a single space.
- Each encoded word is separated from the next by three spaces.
- Each encoded line is rendered as a paragraph, wrapped on word boundaries to prevent the string spanning 80 columns or more when printed.
- Each encoded line is separated from the next by a blank line.

Thus, the first two lines of Lewis Carroll's "Jabberwocky":

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;

would be rendered as:

..... -
-
.....
-
