



Coimisiún na Scrúduithe Stáit
State Examinations Commission

Leaving Certificate Examination 2026

Computer Science

Section C

Higher Level

Wednesday 20 May Morning 11:30 – 12:30

80 marks

Do not hand this up.

This document will not be returned to the
State Examinations Commission.

Instructions

There is one section in this paper.

Section C	Programming	One question	80 marks
Answer all question parts			

Answer **all** parts of the question on your digital device.

Calculators may be used during this section of the examination.

The *Formulae and Tables* booklet cannot be used for this section of the examination.

The superintendent will give you a copy of the *Python Reference Guide*.

Ensure that you save your work regularly.

Save your files using the naming structure described at the beginning of each question part.

If you are unable to get some code to work correctly, you can comment out the code so that you can proceed. The code that has been commented out will be reviewed by the examiner.

Rough work pages are provided at the end of this booklet. Please note that this booklet is not to be handed up and will **not** be reviewed by an examiner.

At the end of the examination it is your responsibility to ensure that you have saved your files onto your external media.

You will be provided with a brown envelope for your external media. Write your examination number on this envelope and place your external media into it before sealing. Place this envelope in the pouch at the front of the red envelope that contains your examination booklet from Section A and B.

There is no examination material on this page

Answer **all** question parts.

Question 16

- (a) Open the program called **Question16_A.py** from your device. The source code is shown below and described briefly on the next page.

Before making any changes, you should save your working copy of the file using the format **ExaminationNumberQuestion16_A.py**. For example, you would save the file as **123456Question16_A.py** if your Examination Number was 123456.

Enter your Examination Number in the space provided on **line 2** in your Python file.

```
1 # Question 16 (a)
2 # Examination Number:
3
4 # Function to calculate readability score of text
5 def calculate_readability_score(text):
6
7     # Split the text into a list of words
8     words = text.split()
9     print("The list of words in the text is:\n", words)
10
11     # Initialise the word counters
12     word_count = len(words) # Number of words in list
13     short_word_count = 0
14
15     # Initialise the number of sentences
16     sentence_count = text.count(".")
17
18     # Calculate the readability score
19     score = word_count * sentence_count
20     score = round(score, 2) # Round the score to 2 d.p.
21
22     return score
23
24 # Set the text for analysis
25 text1 = "Elaborate sentences influence readability in subtle
26 and unpredictable ways."
27 text2 = "I do not like green eggs and ham! I do not like
28 them Sam I am!"
29
30 # Calculate the score for text1
31 score1 = calculate_readability_score(text1)
```

A readability score is an estimate of how easy or difficult it is to read a piece of text.

This program contains a function called `calculate_readability_score` that determines and returns the readability score of a piece of text passed into it. The score is calculated on line 19 by multiplying the number of words (`word_count`) by the number of sentences (`sentence_count`) in the text being processed.

Lines 25 and 26 initialise two variables - `text1` and `text2` - with sentences for the function to process.

Line 29 calls the function `calculate_readability_score` for `text1` and stores the result in `score1`.

When the program is run the output looks as follows:

```
The list of words in the text is:
['Elaborate', 'sentences', 'influence', 'readability',
'in', 'subtle', 'and', 'unpredictable', 'ways.']
```

Make the following changes to the program:

- (i) Modify the program so that it displays the readability score for `text1` followed by a blank line.

When the program is run the output may now look as follows:

```
The list of words in the text is:
['Elaborate', 'sentences', 'influence', 'readability',
'in', 'subtle', 'and', 'unpredictable', 'ways.']
Readability Score 1: 9
```

- (ii) Extend the program so that it calls the function `calculate_readability_score` for `text2` and stores the result in a variable called `score2`. Display the readability score for `text2` followed by a blank line.

When the program is run the output may now look as follows:

```
The list of words in the text is:
['Elaborate', 'sentences', 'influence', 'readability',
'in', 'subtle', 'and', 'unpredictable', 'ways.']
Readability Score 1: 9

The list of words in the text is:
['I', 'do', 'not', 'like', 'green', 'eggs', 'and',
'ham!', 'I', 'do', 'not', 'like', 'them', 'Sam', 'I',
'am!']
Readability Score 2: 0
```

This question continues on the next page.

- (iii) The program uses a full stop to determine a sentence ending. Modify the program so that it also counts exclamation marks (!) as sentence endings.

When the program is run the output may now look as follows:

```
The list of words in the text is:
['Elaborate', 'sentences', 'influence', 'readability',
'in', 'subtle', 'and', 'unpredictable', 'ways.']
Readability Score 1: 9

The list of words in the text is:
['I', 'do', 'not', 'like', 'green', 'eggs', 'and',
'ham!', 'I', 'do', 'not', 'like', 'them', 'Sam', 'I',
'am!']
Readability Score 2: 32
```

- (iv) Extend the program to compare the two scores - score1 and score2 - as follows:
- If the scores are the same display the message: *Both scores are the same.*
 - If score1 is less than score2 display the message:
The first text is easier to read.
 - If score2 is less than score1 display the message:
The second text is easier to read.

When the program is run the output may now look as follows:

```
The list of words in the text is:
['Elaborate', 'sentences', 'influence', 'readability',
'in', 'subtle', 'and', 'unpredictable', 'ways.']
Readability Score 1: 9

The list of words in the text is:
['I', 'do', 'not', 'like', 'green', 'eggs', 'and',
'ham!', 'I', 'do', 'not', 'like', 'them', 'Sam', 'I',
'am!']
Readability Score 2: 32

The first text is easier to read.
```

- (v) Line 13 in the original program initialises a variable called `short_word_count` to 0. A short word is any word that contains two characters or less. Modify the program so that it counts the number of short words in each piece of text. Store the result in `short_word_count` and display its value.

When the program is run the output may now look as follows:

```
The list of words in the text is:
['Elaborate', 'sentences', 'influence', 'readability',
'in', 'subtle', 'and', 'unpredictable', 'ways.']
Short Words: 1
Readability Score 1: 9

The list of words in the text is:
['I', 'do', 'not', 'like', 'green', 'eggs', 'and',
'ham!', 'I', 'do', 'not', 'like', 'them', 'Sam', 'I',
'am!']
Short Words: 5
Readability Score 2: 32

The first text is easier to read.
```

- (vi) Update the code that calculates the readability score to use the formula:

$$0.4 \times \left(\frac{\text{total words}}{\text{total sentences}} + 100 \times \frac{\text{total words} - \text{short words}}{\text{total words}} \right)$$

where,

- total words is the variable `word_count`
- total sentences is the variable `sentence_count`
- short words is the variable `short_word_count`

When the program is run the output may now look as follows:

```
The list of words in the text is:
['Elaborate', 'sentences', 'influence', 'readability',
'in', 'subtle', 'and', 'unpredictable', 'ways.']
Short Words: 1
Readability Score 1: 39.16

The list of words in the text is:
['I', 'do', 'not', 'like', 'green', 'eggs', 'and',
'ham!', 'I', 'do', 'not', 'like', 'them', 'Sam', 'I',
'am!']
Short Words: 5
Readability Score 2: 30.7

The second text is easier to read.
```

This question continues on the next page.

- (vii) Modify the program so that it counts the number of complex words in each sentence. A word is counted as complex if it contains three or more syllables.

A simple heuristic for counting syllables is to count each vowel (a, e, i, o, u) as a syllable and then subtract 1 if the word ends with the letter 'e', known as 'the silent e rule'.

Examples:

- The word *make* has one syllable because it has two vowels but ends in an 'e'.
- The word *readability* has five syllables because it has five vowels (e, a, a, i and i).

When the program is run the output may now look as follows:

```
The list of words in the text is:
['Elaborate', 'sentences', 'influence', 'readability',
'in', 'subtle', 'and', 'unpredictable', 'ways.']
Short Words: 1
Complex Words: 5
Readability Score 1: 39.16

The list of words in the text is:
['I', 'do', 'not', 'like', 'green', 'eggs', 'and',
'ham!', 'I', 'do', 'not', 'like', 'them', 'Sam', 'I',
'am!']
Short Words: 5
Complex Words: 0
Readability Score 2: 30.7

The second text is easier to read.
```


- (b) Open the program called **Question16_B.py** from your device. This file contains only two comments, on lines 1 and 2.



Before adding any code, you should save your working copy of the file using the format **ExaminationNumberQuestion16_B.py**.

For example, you would save the file as **123456Question16_B.py** if your Examination Number was 123456.

Enter your Examination Number in the space provided on **line 2** in your Python file.

You are required to implement a word scoring game.

Example outputs are shown on the next page.

You should use comments throughout your program to explain your code. You may wish to reuse some of the code you used in **part (a)** as part of your solution.

The program should proceed according to the following sequence:

- Ask the user to enter a word using the prompt:
`Enter a word:`
- The word must contain only letters (A-Z or a-z).
If the user enters any digits, spaces or punctuation, the program should display an error message and ask for the word again.
- Convert the word to uppercase before scoring it.
- Score the word according to the following rules:
 - 1 point for each of the letters: A, E, I, O and U
 - 5 points for each of the letters: V, W, X, Y and Z
 - 3 points for all other letters.
- After scoring each word the program should:
 - display the word and its score and the total score so far in an appropriate message
 - keep track of the highest scoring word so far, so that it can be displayed at the end of the game.
- The program should then ask the user if they want to play again using the prompt:
`Play again? [Y/N] :`
If the user enters Y or y, the game is repeated.
If the user enters anything other than Y or y the game ends and the program should display a message with the highest scoring word along with its score.

This question continues on the next page.

Sample output 1:

```
Enter a word: python
Word: PYTHON, Score: 18, Total score: 18
Play again? [Y/N]: y
Enter a word: exam
Word: EXAM, Score: 10, Total score: 28
Play again? [Y/N]: n

Your total score was: 28
The highest scoring word was PYTHON, 18 points
```

Sample output 2:

```
Enter a word: test123
Invalid word.
Enter a word: testing 1 2 3
Invalid word.
Enter a word: test@ing!
Invalid word.
Enter a word: testing
Word: TESTING, Score: 17, Total score: 17
Play again? [Y/N]: Y
Enter a word: Examination
Word: EXAMINATION, Score: 23, Total score: 40
Play again? [Y/N]: Y
Enter a word: computer
Word: COMPUTER, Score: 18, Total score: 58
Play again? [Y/N]: N

Your total score was: 58
The highest scoring word was EXAMINATION, 23 points
```

Space for rough work.

This page will not be reviewed by an examiner.

Do not hand this up.

This document will not be returned to the
State Examinations Commission.

Copyright notice

This examination paper may contain text or images for which the State Examinations Commission is not the copyright owner, and which may have been adapted, for the purpose of assessment, without the authors' prior consent. This examination paper has been prepared in accordance with Section 53(5) of the *Copyright and Related Rights Act, 2000*. Any subsequent use for a purpose other than the intended purpose is not authorised. The Commission does not accept liability for any infringement of third-party rights arising from unauthorised distribution or use of this examination paper.

Leaving Certificate – Higher Level

Computer Science – Section C

Wednesday 20 May

Morning 11:30 – 12:30