



Coimisiún na Scrúduithe Stáit
State Examinations Commission

Leaving Certificate Examination 2023

Computer Science

Section C

Higher Level

Wednesday 24 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.

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.



The program defines a function called `guess_game` which simulates a guessing game.

When the function is called, the program generates a secret number between 1 and 5 inclusive. This is stored in a variable called `secret_number`.

The program then enters a loop in which the user is prompted to guess the number. The user's guess is stored in the variable `user_guess`. Every time the user enters a guess the program increases the variable `guess_count` by one. If the user's guess matches the computer's secret number, the message *'Congratulations! You win!'* is displayed. The loop continues until the user guesses the secret number.

```
1 # Question 16(a)
2 # Examination Number:
3 from random import randint
4
5 def guess_game(max_guesses_allowed):
6
7     secret_number = randint(1, 5)
8     guess_count = 0
9     user_guess = 0
10
11     while (user_guess != secret_number):
12
13         user_guess = int(input("Enter your guess: "))
14         guess_count += 1 #Increase guess_count by 1
15         if user_guess == secret_number:
16             print("Congratulations! You win!")
17
18 print("Welcome to the guessing game!")
19 guess_game(3)
```

This question continues on the next page.

When the program is run, the output may look as follows.
Your output may be different as the secret number is generated randomly.

```
Welcome to the guessing game!  
Enter your guess: 2  
Enter your guess: 3  
Enter your guess: 1  
Congratulations! You win!
```

Make the following changes to the program:

- (i) Change the program to display an extra line of output when the user wins, showing the number of guesses taken.

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

```
Welcome to the guessing game!  
Enter your guess: 2  
Enter your guess: 3  
Enter your guess: 1  
Congratulations! You win!  
You took 3 guesses.
```

- (ii) The program does not display a message unless the user guesses the secret number. Change the program so that it displays one of the following messages as a hint to the user:

'Sorry! Your guess was too low' or 'Sorry! Your guess was too high'

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

```
Welcome to the guessing game!  
Enter your guess: 1  
Sorry! Your guess was too low  
Enter your guess: 5  
Sorry! Your guess was too high  
Enter your guess: 3  
Sorry! Your guess was too low  
Enter your guess: 4  
Congratulations! You win!  
You took 4 guesses.
```

This question continues on the next page.

- (iii) Currently the program has no way of ending unless the user guesses the secret number.
Change the program so that it does not allow the user more than three guesses. This is the value currently being passed into the function.

Hint: You will need to change the loop so that it continues as long as the user's guess is not equal to the secret number and the number of guesses is less than `max_guesses_allowed`.

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

```
Welcome to the guessing game!  
Enter your guess: 1  
Sorry! Your guess was too low  
Enter your guess: 5  
Sorry! Your guess was too high  
Enter your guess: 2  
Sorry! Your guess was too low
```

- (iv) Currently the number of guesses that the user is allowed is hard coded to three.
Modify the program so that the user is presented with the prompt:

Enter the maximum number of guesses allowed.

Store the value entered as an integer and pass this value into the function `guess_game`.

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

```
Welcome to the guessing game!  
Enter the maximum number of guesses allowed: 2  
Enter your guess: 1  
Sorry! Your guess was too low  
Enter your guess: 5  
Sorry! Your guess was too high
```

- (v) Currently the secret number is always between 1 and 5 inclusive.

Modify the program so that it prompts the user to enter a difficulty level – 'E' for easy and 'H' for hard. If the user chooses 'H' the secret number should be between 1 and 100 inclusive. A value of anything other than 'H' can be interpreted as easy.

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

```
Welcome to the guessing game!  
Enter the maximum number of guesses allowed: 3  
Enter difficulty E(asy) or H(ard): H  
Enter your guess: 50  
Sorry! Your guess was too low  
Enter your guess: 75  
Sorry! Your guess was too low  
Enter your guess: 95  
Sorry! Your guess was too low
```

This question continues on the next page.

- (vi) Modify the code so that if the user guesses a number that was already guessed the following message is displayed: *'You already guessed this number.'*

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

```
Welcome to the guessing game!  
Enter the maximum number of guesses allowed: 3  
Enter difficulty E(asy) or H(ard): E  
Enter your guess: 1  
Sorry! Your guess was too low  
Enter your guess: 1  
You already guessed this number.  
Sorry! Your guess was too low  
Enter your guess: 3  
Congratulations! You win!  
You took 3 guesses.
```

Save your file using the format **ExaminationNumberQuestion16_A.py**. For example, you would save the file as **123456Question16_A.py** if your Examination Number was 123456.

This question continues on the next page.

- (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.

Implement a number guessing game in Python. The game requires the user to guess a random number between 1 and 100 and they are awarded points based on how close they get to the number.

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 game should proceed according to the following sequence:

- The user's score is set to zero.
- At the start of every new round the computer generates a random number between 1 and 100. This is the secret number that the user is trying to guess.
- The user is then prompted to enter a guess as shown below.

```
Enter your guess :
```

- Calculate the difference between the secret number and the user guess.
- The numbers are then displayed in a meaningful message on a single line as shown below.

```
Secret number is 55. You guessed 50. Difference is 5.
```

- The numbers are compared and the score is updated according the following rules:
 - If the user's guess and the secret number are the same then the score is increased by 100 and the message "*JACKPOT!!! You score 100 points*" is displayed.
 - If the user's guess is within 20 (either side) of the secret number the user's score is increased by 20 and the message "*You score 20 points*" is displayed.
 - If the user's guess is more than 30 away from the secret number the user's score is decreased by 30 and the message "*You lose 30 points*" is displayed.
- At the end of each round the score is displayed and user is prompted with the message "*Play again? (Y/N):*"
- The game ends when the user enters anything other than the single letter 'Y'.

An example output is shown on the next page.

This question continues on the next page.

Example:

```
Enter your guess: 35
Secret number is 31. You guessed 35. Difference is 4.
You score 20 points
Your total score is: 20
Play again? (Y/N): Y
Enter your guess: 91
Secret number is 6. You guessed 91. Difference is 85.
You lose 30 points
Your total score is: -10
Play again? (Y/N): Y
Enter your guess: 54
Secret number is 54. You guessed 54. Difference is 0.
JACKPOT!!! You score 100 points
Your total score is: 90
Play again? (Y/N): Y
Enter your guess: 50
Secret number is 28. You guessed 50. Difference is 22.
Your total score is: 90
Play again? (Y/N): N
```

Use the format **CandidateNumberQuestion16_B.py** to save your file. For example, you would save the file as **123456Question16_B.py** if your candidate number was 123456.

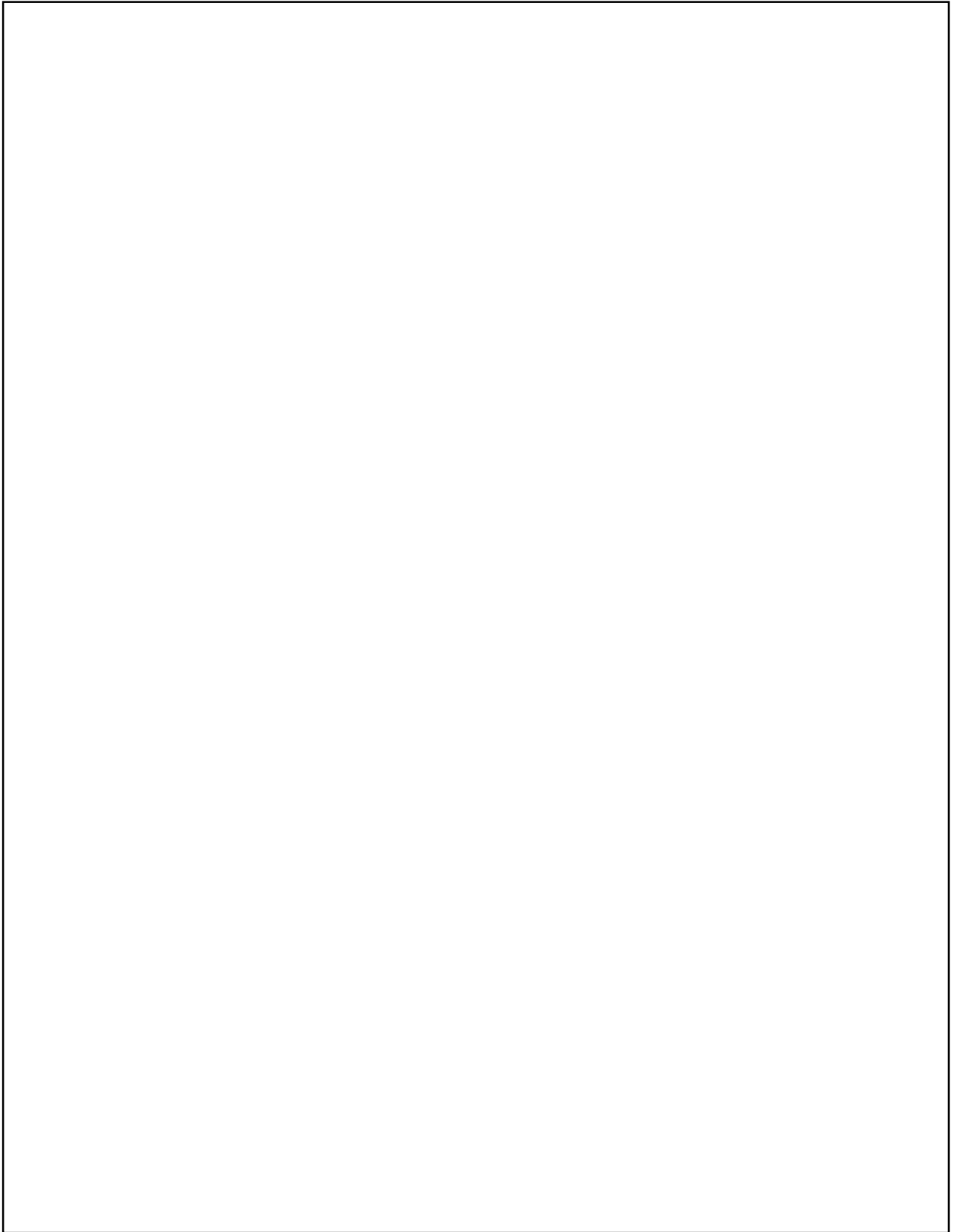
Space for rough work.

This page will not be reviewed by an examiner.

A large, empty rectangular box with a thin black border, occupying most of the page. It is intended for students to use as space for rough work during an examination.

Space for rough work.

This page will not be reviewed by an examiner.



Acknowledgements

Images

Image on page 4 generated by <https://openai.com/dall-e-2/>

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 24 May

Morning 11:30 – 12:30