2024L219A2EL

Coimisiún na Scrúduithe Stáit
State Examinations Commission

Leaving Certificate Examination 2024

# Computer Science

## Section C

## Higher Level

Wednesday 22 May  Morning 11:30 – 12:30

80 marks

Do not hand this up.

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

There is no examination material on this page

## Instructions

There is **one** section in this paper.

| | | | |
|---|---|---|---|
| Section C | Programming | **One** question<br>Answer **all** question parts | 80 marks |

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.

| Section C | Programming | 80 marks |
|---|---|---|

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 initialises a list called `fruits` with three elements – *apple*, *cherry* and *orange*. Line 7 of the program is an assignment statement in which a random fruit is selected from the list and assigned to the variable called `random_fruit_1`.
The program does not display any output.

```
1  # Question 16(a)
2  # Examination Number:
3  from random import choice
4
5  fruits = ['apple', 'cherry', 'orange']
6
7  random_fruit_1 = choice(fruits)
```

Make the following changes to the program:

**(i)** Write a line of code to display the value of the variable `random_fruit_1` in a message.

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

```
Random Fruit 1: cherry
```

**(ii)** Add statements to initialise **two** new variables with fruits chosen randomly from the list. You should also display the values of the variables which should be called `random_fruit_2` and `random_fruit_3`.

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

```
Random Fruit 1: orange
Random Fruit 2: orange
Random Fruit 3: cherry
```

**(iii)** Add code to display the message "First fruit is cherry" if the first random fruit is a cherry.

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

```
Random Fruit 1: cherry
Random Fruit 2: apple
Random Fruit 3: cherry

First fruit is cherry
```

**(iv)** Add code to display the message "First pair match" if the first two fruits are the same.

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

```
Random Fruit 1: cherry
Random Fruit 2: cherry
Random Fruit 3: apple

First fruit is cherry
First pair match
```

**(v)** Add code to display the message "First pair are cherries" if the first two fruits are both cherries.

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

```
Random Fruit 1: cherry
Random Fruit 2: cherry
Random Fruit 3: apple

First fruit is cherry
First pair match
First pair are cherries
```

**(vi)** Add code to display the message "Matching pair" if any two fruits are the same.

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

```
Random Fruit 1: apple
Random Fruit 2: cherry
Random Fruit 3: apple

Matching pair
```

*This question continues on the next page.*

**(vii)** Extend the program with a loop that iterates 100 times. The loop should generate a random fruit on each iteration. After the loop is executed, the program should display a count of the number of times each fruit was generated. There is no need to display the names of the 100 fruits.

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

```
Random Fruit 1: orange
Random Fruit 2: cherry
Random Fruit 3: cherry

Matching pair

apple 33
cherry 36
orange 31
```

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.

**(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 simulation of a fruit machine in Python.
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:

- Initialise a list called `fruits` with three elements – *apple*, *cherry* and *orange*.
- Display the initial list of fruits as shown.

```
The initial list of fruits is:
['apple', 'cherry', 'orange']
```

- Prompt the user to enter an additional fruit, for example kiwi, pear or lemon and append the value entered to `fruits`.

```
Enter an additional fruit: kiwi
```

- Display the list of four fruits, for example:

```
The list of four fruits is:
['apple', 'cherry', 'orange', 'kiwi']
```

- Prompt the user to nominate their winning fruit which must be in the above list. If the user enters a fruit that is not in `fruits`, the program should display an error message and prompt the user to nominate their winning fruit again. This should continue as long as the winning fruit entered is not in `fruits`.

```
Nominate your winning fruit: cherry
```

- Display the winning fruit, as show below.

```
Nominate your winning fruit: cherry
The winning fruit you selected is cherry
```

- Write code to select three random fruits from `fruits`. Compare the selected fruits to the winning fruit entered earlier and keep going until all three fruits match the winning fruit.
  The program should keep a count of the number of tries taken and display this with a "Winner" message at the end, as shown below.

```
Winner after 38 tries
```

*This question continues on the next page.*

Two example outputs are shown below.

**Sample output 1**:

```
The initial list of fruits is:
['apple', 'cherry', 'orange']

Enter an additional fruit: kiwi
The list of 4 fruits is:
['apple', 'cherry', 'orange', 'kiwi']

Nominate your winning fruit: cherry
The winning fruit you selected is cherry

Winner after 38 tries
```

**Sample output 2:**

```
The initial list of fruits is:
['apple', 'cherry', 'orange']

Enter an additional fruit: kiwi
The list of 4 random fruits is:
['apple', 'cherry', 'orange', 'kiwi']

Nominate your winning fruit: pear
Error: winning fruit must be in the list
Nominate your winning fruit: orange

The winning fruit you selected is orange

Winner after 27 tries
```

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.

Space for rough work.

This page will not be reviewed by an examiner.

Space for rough work.

This page will not be reviewed by an examiner.

**Acknowledgements**

**Images**
Image on page 4: https://www.flaticon.com/free-icons/fruit-machine
Image on page 7: https://dm0qx8t0i9gc9.cloudfront.net/thumbnails/video/qEue9C6/videoblocks-slot-machine-wheels-three-3-cherries-jackpot-winner-3-d-animation_bkbn1fknu_thumbnail-1080_09.png

Do not hand this up.

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



Leaving Certificate – Higher Level

# Computer Science – Section C

Wednesday 22 May
Morning 11:30 – 12:30