

Pre-Leaving Certificate Examination, 2023

Computer Science

Section C

Higher Level

Time: 1 hour

80 marks

Do not hand this question paper up.
This document will not be examined
and it will not be returned to you.

Instructions

There is one section in this paper.

Section C	Programming	One question 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 all of your files onto your external media.

Answer all question parts.

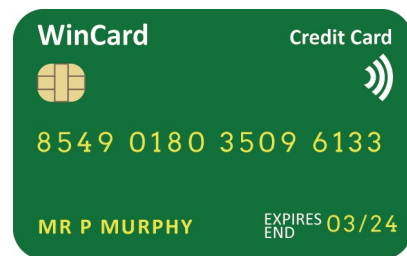
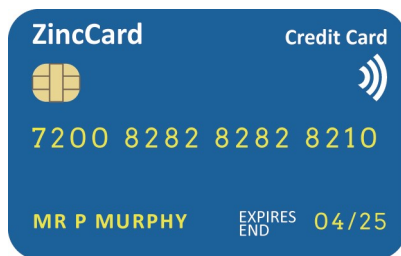
Question 16

- (a) Open the program called **Question16_A.py** from your device.

Before making any changes, you should save your working copy of the file using the format **StudentNameQuestion16_A.py**. For example, you would save the file as **PatMurphyQuestion16_A.py** if your name was Pat Murphy.

Enter your Name and School in the space provided on **line 2** in your Python file.

Validating credit cards is an important tool for FinTech companies. Shown below are credit cards offered by two fictional companies, ZincCard and WinCard:



```

1 # Question 16(a)
2 # Name and School:
3
4 cardNum=7200828282828210
5
6 print(cardNum)
```

Make the following changes to the program:

- (i) Amend the file so that the output reads as follows:

```
Welcome to CardCheck. Enter your card number:
7200828282828210
```

- (ii) ZincCard numbers begin with a 7 and WinCard numbers begin with an 8. Modify the program so that it detects whether the card number is for a ZincCard or a WinCard. (You can use 7200828282828210 to validate that it is a ZincCard.)

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

```
Welcome to CardCheck. Enter your card number:
7200828282828210
This is a ZincCard
```

- (iii) Amend the file so that it can accept any ZincCard or WinCard number.
- (iv) Both credit cards use a 16-digit number. Modify the program to check that the length of the card number is correct. If it isn't, the program should prompt the user to re-enter the card number. If the user's **initial input and two other attempts** fail, the user is "blocked" and the program stops.

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

```
Welcome to CardCheck. Enter your card number: 8
That is incorrect, please try again: 7
That is incorrect, please try again: 7200828282828210
That is correct
This is a ZincCard
```

- (v) For extra security, a card verification value (CVV) 3-digit number is used in conjunction with the 16 digit number and the expiry date of the card.

To generate a unique CVV

- Add up the digits in the card expiry date, e.g. 04/25 = $0 + 4 + 2 + 5 = 11$
- Multiply this by the first 2 digits in the card number, e.g. for ZincCard = 72
- Subtract the 10th digit of the card, e.g. for ZincCard = 2
- e.g. $11 \times 72 - 2 = 790$

Modify the program so that it outputs the correct CVV for the relevant card number.

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

```
Welcome to CardCheck. Enter your card number:
7200828282828210
Enter the card expiry date e.g. 11/26 should be entered
as 1126: 0425
This is a ZincCard
CVV number: 790
```

- (vi) Place a single comment in your code where you have had to use any Boolean notation.

- (vii) Modify the program so that the output looks as follows:

```
Welcome to CardCheck. Enter your card number:
8549018035096133
Enter the card expiry date e.g. 11/26 should be entered
as 1126: 0324
This is a WinCard
CVV number: 760
Card number: 8549-0180-3509-6133 and it is valid
```

Save your file using the format **StudentNameQuestion16_A.py**. For example, you would save the file as **PatMurphyQuestion16_A.py** if your name was Pat Murphy.

- (b) Open the program called **Question16_B.py** from your device.

Before making any changes, you should save your working copy of the file using the format **StudentNameQuestion16_B.py**. For example, you would save the file as **PatMurphyQuestion16_B.py** if your name was Pat Murphy.

Enter your Name and School in the space provided on **line 2** in your Python file.

The Luhn algorithm, also known as the **modulus 10** or **mod 10** algorithm, is a simple checksum formula used to validate a variety of identification numbers, such as credit card numbers.

Steps in the Luhn algorithm:

1. Remove the last digit from the card number. This number is called the checking digit and will be used at step 4.
2. Reverse the order of the remaining digits.
3. For this sequence of reversed digits, take the digits at **each of the even indices** (0, 2, 4, 6, etc.) and double them. If a result is greater than 9, subtract 9 from that number.
4. Add together all of the results and add the checking digit removed at step 1.
5. If the result is divisible by 10, the number is a valid card number. If it is not, the card number is not valid.

Here is an example using WinCard number 8549018035096133:

1.	8	5	4	9	0	1	8	0	3	5	0	9	6	1	3	X
2.	3	1	6	9	0	5	3	0	8	1	0	9	4	5	8	X
3.	3	1	6	9	0	5	3	0	8	1	0	9	4	5	8	X
	↓		↓		↓		↓		↓		↓		↓		↓	X
	6		12 – 9		0		6		16 – 9		0		8		16 – 9	

4. $6 + 1 + 3 + 9 + 0 + 5 + 6 + 0 + 7 + 1 + 0 + 9 + 8 + 5 + 7 + 3 = 70$
5. $70 \div 10 = 7$ so this card is valid.

Write a program that validates either of the credit card numbers used in **question 16(a)**.

Save your file using the format **StudentNameQuestion16_B.py**. For example, you would save the file as **PatMurphyQuestion16_B.py** if your name was Pat Murphy.

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

Images on page 3: DEB Exams

Do not hand this question paper up.

This document will not be examined
and it will not be returned to you.

Copyright notice

This examination paper may contain text or images for which DEB Exams 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. DEB Exams does not accept liability for any infringement of third-party rights arising from unauthorised distribution or use of this examination paper.

Pre-Leaving Certificate Examination, 2023 – Higher Level

Computer Science – Section C

Time: 1 hour