



# CFGDEGREE

## FOUNDATION EXAM - TRACK 1 - 1.5 HOURS

SECTION	MARK
Multiple Choice Questions	10
Code/Query Evaluation	15
Coding Challenge	25
Total	50

### Important notes:

- Please make sure you have read the **CFGdegree Exam Submission Instructions** prior to starting this paper (these have been shared via your Slack channel).
  - You have 1 hour and 30 minutes to complete this paper, unless you have been granted additional time by the programmes team in which case you will have 2 hours to complete this paper.
  - This is an open book exam. You may be required to use your existing knowledge to search for additional information or new functions.
  - You should use the relevant software for the papers you have chosen (e.g. Pycharm, MySQL, VS Code.) If you are using CodePen, your work must be downloaded prior to being submitted as a file via GitHub. Links to live CodePens will **not** be accepted as final submissions.
  - Submission must be made via GitHub within 30 minutes of the exam finishing.
-

## Section One: Multiple Choice Questions (10 marks)

### Question 1:

What is the output of the following python equation:  $5 + 3 / 2$ ?

- A. 6.5
- B. 4
- C. 4.0
- D. 7

### Question 2:

What would be the first line and last line output of the following python code?

```
for i in range(10):  
    print("o" * i)
```

- A. First line: o  
Last line: oooooooooo
- B. First line: 0  
Last line: 0000000000
- C. First line:  
Last line: oooooooooo
- D. First line: o  
Last line: oo

### Question 3:

Which of these is a necessary feature of an efficient loop?

- A. Iterates through a string
- B. Works in ascending order
- C. Return statement
- D. Exit condition

#### Question 4:

Which statement is **not** true about the functionality of Python's return statement?

- A. It exits the function
- B. By default it gives a None value
- C. It prints a message to the console
- D. It gives a value back to where it was called

#### Question 5:

Which of the following adds a string from a list of courses, to a new list if the string contains "cfg"

- A.  

```
for course in courses:  
    if course.contains("cfg"):  
        courses.append(course)
```
- B.  

```
new_list = [x for x in courses if "cfg" in x]
```
- C.  

```
new_list = {x for x in courses if "cfg" in x}
```
- D.  

```
new_list = [ ]  
for course in courses:  
    if course.contains("cfg"):  
        new_list.append(course)
```

#### Question 6:

What does the SQL statement "SELECT COUNT(DISTINCT column\_name) FROM table\_name;" do?

- A. Counts the total number of rows in the table
- B. Counts the number of unique values in the specified column
- C. Retrieves the average value of the specified column
- D. Returns the sum of all values in the specified column

**Question 7:**

Which SQL operator is used to retrieve rows that exist in the first table but not in the second table in a JOIN operation?

- A. INNER JOIN
- B. LEFT JOIN
- C. RIGHT JOIN
- D. OUTER JOIN

**Question 8:**

A company wants to retrieve all employees that work in the sales department where their salary is greater than £50,000. Which SQL query can achieve this?

- A. `SELECT * FROM employees WHERE department = "sales" AND salary > 50000;`
- B. `SELECT * FROM employees WHERE department == "sales" AND salary > 50000;`
- C. `SELECT * FROM employees WHERE department = "sales" OR salary > 50000;`
- D. `SELECT * FROM employees WHERE department = "sales" OR salary < 50000;`

***QUESTIONS CONTINUED ON THE NEXT PAGE***

---

### Question 9:

Look at the following tables: "students" and "courses."

Students Table:

student_id	student_name
1	John Smith
2	Jane Doe

Courses Table:

course_id	course_name	student_id
101	Chemistry	1
102	English	1

Which SQL query is used to retrieve the names of students who have **not** enrolled in any courses?

- A. SELECT student\_name FROM students;
- B. SELECT student\_name FROM students LEFT JOIN courses ON students.student\_id = courses.student\_id WHERE courses.course\_id IS NULL;
- C. SELECT student\_name FROM students INNER JOIN courses ON students.student\_id = courses.student\_id WHERE courses.course\_id IS NULL;
- D. SELECT student\_name FROM students JOIN courses ON students.student\_id = courses.student\_id WHERE courses.course\_id IS NULL;

**QUESTIONS CONTINUED ON THE NEXT PAGE**

---

**Question 10:**

Review the table named "employees" with columns "employee\_id" and "salary."

employee_id	salary
1	50000
2	60000
3	47000
4	22000
5	70000

Which SQL query can be used to find the lowest salary among employees with salaries greater than \$50,000 in the "employees" table?

- A. `SELECT MIN(salary) FROM employees WHERE salary > 50000;`
- B. `SELECT DISTINCT salary FROM employees WHERE salary > 50000 ORDER BY salary DESC LIMIT 1;`
- C. `SELECT salary FROM employees WHERE salary < 50000 ORDER BY salary ASC LIMIT 1;`
- D. `SELECT TOP 1 salary FROM employees WHERE salary > 50000 ORDER BY salary ASC;`

***QUESTIONS CONTINUED ON THE NEXT PAGE***

---

## Section Two: Query Evaluation (15 marks)

Consider a scenario where a healthcare authority aims to view the patients' data from the Cardiology Department, which currently comprises approximately 8500 admitted patients. The objective is to identify patients with the highest number of admissions from January 2015 onwards.

The existing query is as follows:

```
1  SELECT
2      patient_id,
3      COUNT(*) AS admission_count
4  FROM
5      admissions
6  WHERE
7      department_id = (SELECT department_id FROM departments WHERE department_name = 'Cardiology'
8      OR department_name = NULL)
9      AND admission_date >= '2015-01-01'
10 GROUP BY
11     patient_id
12 ORDER BY
13     admission_count DESC
14 LIMIT 1;
```

Evaluate the given code. You should:

1. Identify and correct any errors in the code - include any corrected code snippets alongside your written answer,
2. Provide **three** specific suggestions for improving or optimising the code.

*Please note this is an evaluation task and you do **not** need to rewrite the entire section of code or run the code to get the output.*

**QUESTIONS CONTINUED ON THE NEXT PAGE**

---

## Section Three: Coding Challenge (25 marks)

You have been tasked with creating and testing a function that can solve the handshake problem.

### The Handshake Problem

The handshake problem describes the calculation of how many unique handshakes can occur within a group of  $X$  people.

For a group of 2 people this is 1 handshake:  $Person\ A \Rightarrow Person\ B$

For a group of 3 people this is 3 handshakes:

1.  $Person\ A \Rightarrow Person\ B$
2.  $Person\ A \Rightarrow Person\ C$
3.  $Person\ B \Rightarrow Person\ C$

But for a bigger number like 20, this results in 190 handshakes.

### The Challenge

You will need to:

- Write a function that takes in an integer value for the number of people and returns an integer with the number of handshakes
  - HINT: There is a formula online for this problem that generalises the equation
- Validate if a handshake can occur given  $X$  as an input
- Identify an error state and throw a custom exception
- Create a test file for the function and create a comprehensive test suite

To start, use the skeleton code provided in the **exam\_python.py** file.

You can add more helper functions and code structures as you see fit.

---

**END OF PAPER**