

# OCR GCSE Computer Science (9-1) - J276 (from 2016)

## Component 3: Non-Exam Assessment (NEA) (J276/03)

### NEA Support Pack



# Contents

## NEA Support Pack

NEA Procedures - Introduction for students	4
Technical Preparation for the NEA component	7
<b>Help sheet 1</b> - Creating a board game using Python - Includes a complete set of instructions	8
<b>Help sheet 2</b> - Creating a database using Python - Includes a complete set of instructions	15
<b>Help Sheet 3</b> - Menus, Lists and Exporting using Python - Includes a complete set of instructions	19

## J276 Programming project Sample 1 - Archside Academy Detention Database

NEA Sample project scenario	28
NEA Sample student solution	31
NEA Sample Python code	48

## J276 Programming project Sample 2 - Quiz Board Game

NEA Sample project scenario	52
NEA Sample student solution	55
NEA Sample Python code	71

## Give your students a better chance of success

Edulito is a UK based educational publishing company that has been set up to ensure that students have a real opportunity to fulfil their potential. All of the available resources have been tested in UK schools to ensure that they provide students with a meaningful experience and provide teachers, parents and students with a clear indication of progress.

This support pack is designed to provide a stimulating, engaging and effective way of preparing both teachers and students for the Non-Examined Assessment Component of the 9-1 GCSE in computer science.

The pack contains:

Introduction for students - NEA Procedures

Technical Preparation for the NEA component

Help sheet 1 - Creating a board game using Python - Includes a complete set of instructions

Help sheet 2 - Creating a database using Python - Includes a complete set of instructions

Help Sheet 3 - Menus, Lists and Exporting using Python - Includes a complete set of instructions

J276 Programming project Sample Task 1 - Archside Academy Detention Database.

Including:

- NEA sample project scenario
- NEA sample student solution
- NEA sample Python code

J276 Programming project Sample Task 2 - Quiz Board Game

Including:

- NEA sample project scenario
- NEA sample student solution
- NEA sample Python code

## Terms and Conditions of Use

**Your school has permission to copy this resource as many times as you require and to use it as you wish within your school/organisation.**

**You do not have permission to distribute it as a paper or electronic document to other schools or organisations.**

## Introduction for students - NEA Procedures

### What do you have to do?

- You must think computationally to solve a task and while doing so create a report detailing the creation of your solution, explaining what you did and why you did it.
- OCR will issue a choice of three assessment tasks at the start of year 11.
- The tasks will provide opportunities for you to demonstrate your practical ability in the skills outlined in the specification.
- You must complete **ONE** task.
- You will need to create suitable algorithms which will provide a solution to the problems identified in the task.
- You will then code these solutions in Python.
- The solutions must be tested at each stage to ensure they solve the problem.
- You must use a suitable test plan with appropriate test data.
- The code must be annotated to describe the process.
- Test results should be annotated to show how these relate to the code, the test plan and the original problem.
- You will need to provide an evaluation of your solution based on the test evidence.

### The non-exam assessment requirements.

- You are **not** allowed access to the internet within the non-exam assessment controlled environment.
- All work submitted must have been done under observation by your teacher and the final report must be your own work.
- External sources can be used **but must be referenced** and no marks can be awarded for materials submitted which are not the learner's own.
- When your work is moderated no credit will be given for common coded solutions.
- The non-exam assessment should take a total of **20 hours** to complete.

### The Process

The project can be carried out in many ways but is best approached using an iterative process for developing solutions to the task such as below:

- Success criteria (what will a successful solution be)
- Planning and design (flow charts and pseudocode)
- Development (narrative of the process with explanations of code)
- Testing and remedial actions (with narrative of changes made)
- Evaluation (clearly linked to success criteria).

## Required programming techniques

- how to identify and use variables, operators, inputs, outputs and assignments
- how to understand and use the three basic programming constructs used to control the flow of a program: Sequence; Selection; Iteration
- how to understand and use suitable loops including count and condition controlled loops
- how to use different types of data, including Boolean, string, integer and real, appropriately in solutions to problems
- how to understand and use basic string manipulation
- how to understand and use basic file handling operations:
  - open
  - read
  - write
  - close
- how to define and use arrays (Lists) as appropriate when solving problems
- how to understand and use functions/subprograms to create structured code.

## What can teachers do?

- explain the task
- advise on resources
- provide the support described within the 'Permitted Support' section of the Specification
- question you to ensure that the work is your own
- provide you with a copy of the mark scheme

## What teachers can't do

Teachers must not:

- give **detailed** advice and suggestions as to how the work may be improved in order to meet the assessment criteria. This includes indicating errors or omissions and personally intervening to improve
- the presentation or content of the work
- practise the task with you
- practise tasks which are similar in nature with you
- provide templates, model answers or feedback on drafts
- produce templates or model answers and publish them online.

## Teachers must ensure that:

- you do not access the internet
- you are not allowed to take the NEA tasks home with you
- all work presented for submission must have been completed under supervised conditions
- accounts associated with the NEA tasks must be locked between sessions to ensure that you cannot access them outside of the supervised conditions

- you must not have access to online file storage accounts or email during the supervised conditions in order to prevent you from completing work at home and bringing it into the supervised conditions.

## Presenting the Project

You must observe the following procedures when producing your final piece of work for the project task:

- any copied material must be suitably acknowledged
- work submitted for moderation or marking must be marked with the:
  - centre number
  - centre name
  - learner number
  - learner name
  - component code
  - assignment title.
- The report should be submitted as a single document (.pdf format preferably) and all the code must be visible in the report and be fully annotated.

## Technical Preparation for the NEA component

### Setting up secure accounts

NEA accounts must be totally secure, meaning that candidates can only access their account when under formal control conditions. This should be taken into account when creating NEA accounts for candidate use.

It is the centre's responsibility to ensure that all NEA accounts adhere to the requirements of the specification and JCQ Instructions for NEA.

### Separate to usual school accounts

By creating unique accounts, you may add restrictions to these accounts, such as restriction of internet access, without affecting their day to day study. You may also create unique access to network drives for storage of work, so that there is no way they can access their work outside of formal control conditions.

### Time bound

It is imperative that candidates cannot gain access to these accounts when say, for instance, they go to the library at lunch. Network management programs will allow you to either enable/ disable accounts manually, and most are likely to be able to create a schedule for this.

### Remove potential for external access

Be wary if you are using a Virtual Learning Environment (VLE) or similar. Ensure your NEA accounts cannot be accessed externally by candidates through any portal system you may have within the school, such as a VLE. Also remove any other programs that may allow candidates external access, e.g. Mail Clients, File Sharing, Web Browsers etc.

### Remove shared drive access

It is important that candidates only have access to the resource bank created for that exam series which includes their preparatory notes. Any other access to network locations may allow candidates access to unauthorised materials, or the ability to export/share work with each other.

## Help sheet 1

### Creating a board game using Python

#### 1. Creating a game board

A simple way to create a game board is using a list with every number from the game board. If you want a board like the one below you will need 4 numbers.

4	3
1	2

Copy this code into Python:

```
board=[1,2,3,4]
print(board[3],board[2])
print(board[0],board[1])
```

Remember the positions in a list start from 0, and so number 1 in the list is found in the position 0 and number two in the position 1.

The output from this program is:

```
4 3
1 2
```

You can use print statements to make it more like a game board. Add this to your Python code.

```
board=[1,2,3,4]
print(".....")
print(board[3],":",board[2])
print(".....")
print(board[0],":",board[1])
print(".....")
```

The output from this program is:

```
.....
4 : 3
.....
1 : 2
.....
```

## Scenario

Rob works in the admin department at Archside Academy. He has been asked by the headteacher to create a secure database system to manage detentions within the school. When a detention is set a unique number is used to identify it.

Rob wants to be able to have a user friendly interface that allows him to log into the system and carry out the necessary administration.

The detention system must store the following information:

- Unique number (set by the program)
- Student name
- Form group
- Date of detention
- Time of detention
- Length of detention
- Reason for detention
- Teacher Initials

Analyse the requirements for this system and design, develop, test and evaluate a program that allows Rob to:

1. Log in with a username and password
2. Access a menu system
3. Enter and store detention information
4. Log out
5. Retrieve and display the details of any detentions by searching for a student's name.\*
6. Create at least two different reports that Rob might need, and describe how he would use each one.
7. Produce these reports when selected from a menu.

\*Note for candidates:

In order to test this program, you will need a data file containing the details of at least 20 detentions.