

EDULiTO

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

NEA Support Pack



Photocopiable Resources

Contents

NEA Support Pack

NEA Procedures - Introduction for students	4
Technical Preparation for the NEA component	7
Help sheet 1 - Creating a board game using Python - Includes a complete set of instructions	8
Help sheet 2 - Creating a database using Python - Includes a complete set of instructions	15
Help Sheet 3 - 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 you must create a report detailing the creation of your solution, explaining what you did and why you did it.
- OCR will issue a choice to you a set 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 in computer science.
- You must complete **ONE** task.
- You will need to create suitable algorithms which will provide a solution to the problems identified in the task that you choose.
- You will then code these solutions in Python, or a language of your choice.
- The solutions must be tested at each stage to ensure that they solve the problem.
- You must use a suitable test plan with appropriate test data.
- The code must be annotated to describe the processes used.
- 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 the work submitted by you must have been done under observation by your teacher and the final report must be completely your own work.
- External sources can be used **but must be referenced** and no marks can be awarded for materials submitted which are not your own.
- When your work is marked 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 different ways but it 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 (a narrative of the process with explanations of the code that you have used)
- Testing and remedial actions carried out (with a narrative of the changes made)
- Evaluation (clearly linked to success criteria).

Required programming techniques

- You can identify and use variables, operators, inputs, outputs and assignments
- You can show that you understand and can use the three basic programming constructs used to control the flow of a program i.e. Sequence, Selection and Iteration
- You understand and can use suitable loops including count and condition controlled loops
- You can use different types of data, including Boolean, string, integer and real, appropriately in solutions to the problems you have been set.
- You understand and can use basic string manipulation
- You understand and can use basic file handling operations, including:
 - open
 - read
 - write
 - close
- You can define and use arrays (Lists) appropriately when solving problems
- You understand and can use functions/sub-programs to create structured code.

What can teachers do?

- They can explain the task
- They can advise on resources
- They can provide the support described within the 'Permitted Support' section of the Specification
- They can question you to ensure that the work is your own
- They can 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 your work
- Tell you how to improve the presentation or content of your work
- Practise the task with you
- Practise tasks which are similar in nature with you
- Provide you with 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 do not take the NEA tasks home with you
- All work presented for submission has been completed under supervised conditions
- Logins associated with the NEA tasks are locked between sessions

- You do not have access to online file storage accounts or email during the supervised conditions, 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 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 you produce 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

You must create a unique login for each candidate that only the individual candidate has access to. These accounts must not be accessible outside of the NEA formal conditions. You must ensure that your centre complies with OCR and JCQ Instructions for the NEA.

Separate to usual school accounts

The accounts that have been created by the school must not allow access to the Internet but must provide candidates with a location/drive that can be used to save their work as required during the NEA sessions.

Time bound

The accounts that have been created must be disabled either manually or automatically so that they are not accessible outside of an NEA session,

Remove potential for external access

You must ensure that the NEA is not available to students via the school VLE or any other cloud services. Programs such as mail clients, file sharing services, web browsers etc. should also be removed from the NEA accounts.

Remove shared drive access

Candidates must have access to a location/drive to store their NEA work and another location/drive that contains the NEA Resource bank. The location/drive containing the NEA resource bank must be read-only. They must not have access to any other locations/drives.

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.