

Practice NEA 1

When you start Year 11, you will have to complete a piece of work called Non-Examined Assessment (NEA). This uses the programming skills that you have learnt to solve a particular problem that is set by the exam board. You will have 20 hours to complete the task.

In year 9 and year 10 we will have a practice task. This is designed to give you practice in how to do an NEA when you have to do the real thing.

For the first task you will have 8 lessons.

Secret Spy Code Task

All through history, governments have wanted to keep their messages secret. To make a message secret you can use a number of different *ciphers*. A cipher is a way of scrambling up a message (the *plaintext*) into a jumbled pattern (the *ciphertext*). Nowadays this is done using computers.

Your task is to write a computer program that will *encipher* a message using a Caesar cipher.

A Caesar cipher: this is a very simple cipher where each letter is shifted a certain number of characters along the alphabet. A Caesar cipher using the number 3 will turn A into D, B into E, C into F, and so on. When you get to the end of the alphabet you need to go round and back to the beginning so W turns into Z, X into A, Y into B and Z into C.

A message like I AM NOW HERE will turn into L DO QRZ KHUH

I	A	M	N	O	W	H	E	R	E
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
L	D	O	Q	R	Z	K	H	U	H

To make it less obvious that the message is made up of specific words, the ciphertext is then broken up into groups of 5 letters so I AM NOW HERE will be enciphered as:

LDOQR ZKHUH

Your Task

You are to write a program that will:

1. Show a menu on the screen which displays these options:
 - A. Enter plaintext
 - B. Enter step size
 - C. Convert to ciphertext
 - D. Convert to groups of five
 - Q. QuitEnter your option:
2. If the user types "Q", the program gives a suitable message and quits.
3. If the user enters an **option** that is not A, B, C, D or Q they should be told it is incorrect and shown the menu again.
4. If the user types "A", they are asked to type in a message in capital letters which is stored in a variable called **plaintext**.
5. If the user types "B", they are asked to type in a number between 1 and 25. If they type in a number that is outside this range they are asked to type it again. This number is stored in a variable called **step**.
6. If the user types "C", the program:
 - a. checks that they have entered **plaintext** and **step**; if they haven't, an error message is given and they are sent back to the menu.
 - b. displays the plaintext and the step on the screen,
 - c. converts each letter in **plaintext** into its ASCII number,
 - d. if the ASCII value is not 32 (a space), adds **step** to it,
 - e. if the result is bigger than 90 (ASCII for "Z"), subtracts 26,
 - f. adds each letter to a new string called **ciphertext**,
 - g. displays the ciphertext on the screen.
7. If the user types "D", the program:
 - a. checks that **ciphertext** has been created; if it has not, an error message is given and they are sent back to the menu.
 - b. breaks up the ciphertext into groups of five letters separated by spaces,
 - c. stores the new ciphertext string in a variable called **groupedciphertext**,
 - d. displays the grouped ciphertext on the screen.

Design

1. Create a folder called NEA Practice. In this folder make three subfolders: Design, Implementation and Testing.
2. Start a new Word document called Design.
3. In your Design document, write a summary of what your program must do.
4. Make a table to break the whole task up into subtasks. For each subtask:
 - a. Give it a name (no spaces)
 - b. Describe it
 - c. Say what data it will need
 - d. Say what data it will give back

It will look like this:

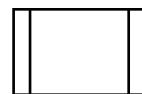
Name	Description	Data needed	Data returned
Menu	Show menu and get chosen option. If not A, B, C, D or Q ask again.	None	option
Quit	Show quit message	None	None

5. Make a table to list the variables that will be in the program. Read the task and you'll see some variables in bold. For each variable say what its purpose and data type is.

Variable Name	Purpose	Data Type
option	Shows which option has been chosen by the user	String

6. Draw a flowchart that shows how the program will work.
 - a. Make one main flowchart that shows each subtask as a single box.
 - b. Make a separate flowchart for each subtask.

To show a subtask in a flowchart use the symbol:



Implementation

Create the program. Check that each part works correctly. You will be graded as follows:

Progress	Grade
Complete task 1	1
Complete tasks 2 and 3	2
Complete tasks 4 and 5	3
Complete task 6	4
Complete task 7	5

Testing

You should be testing each aspect of your program as you go along. This is called *functional testing*.

At the end of the project you should carry out a full test which proves that the whole program works as it should. This is called *system testing*.

Make a table like this and work your way through all the things that the program should do:

Test Number	Test	Type of test (Valid, Erroneous, Boundary)	What should happen	What did happen	Evidence	Notes e.g. any corrections needed
1	Display menu	Valid	Menu displays at beginning	Works correctly	Screenshot 1	-
2	Input Z as option	Erroneous	Error message displays	Menu showed again but no error message	Screenshots 2 and 3	Added new code to display error message

After you have done all the testing, write an evaluation that:

1. Describes how well your program meets the needs of the problem set out at the beginning.
2. Give some ideas about how the program could be improved by:
 - a. fixing problems,
 - b. adding more functionality.