

Level 3 Diploma in Computing

Ofqual Number: 600/6407/9 Quartz ID: 1155

Level 3 International Foundation Diploma for Higher Education Studies

Ofqual Number: 603/0080/2 Quartz ID: 1125

Paper ID: 2177-1

Introduction to Programming with Python

Global Assignment

Title: Interactive game with items

Important notes:

- Please refer to the *Assignment Presentation Requirements* for advice on how to set out your assignment. These can be found on the NCC Education website. Hover over 'About Us' on the main menu and then navigate to 'Policies and Procedures' then scroll to the 'Student Support' area.
- You **must** read the NCC Education document *Academic Misconduct Policy* and ensure that you acknowledge all the sources that you use in your work. These documents are available on the NCC Education website. Hover over 'About Us' on the main menu and then navigate to 'Policies and Procedures' then scroll to the 'Student Support' area.
- You **must** complete the *Statement and Confirmation of Own Work*. The form is available on the NCC Education website. Hover over 'About Us' on the main menu and then navigate to 'Policies and Procedures' then scroll to the 'Student Support' area.
- **Please note, word count is not applicable on this assignment as the artefact to be produced is a programming code (with a word document containing screenshots of the results, a filled-out object definition sheet and a test log)**
- You **must** submit a paper copy and digital copy (on disk or similarly acceptable medium). Media containing viruses, or media that cannot be run directly, will result in a fail grade being awarded for this assessment.
- All electronic media will be checked for plagiarism.

Introduction

Plan, create and test a Python program to meet the given scenario.

You need to produce a working Python program and a Word document containing the following:

- Success criteria.
- Decomposition of the problem.
- Design of modules and algorithms using pseudocode.
- The Python program code.
- Screenshots of the Python program working.
- A test log including a range of tests and screenshot evidence of the results.
- Installation and user guide documentation.

Scenario

A computer program needs developing that allows user to navigate a simulated world.

The program has the following requirements:

- Single player game, the user can enter the name of their character.
- The player has a bag where they can collect and store up to 4 items.
- The game outputs the current position of the player and the choices they can make.
Some **example** choices are:
 - Do you want to move forward or to the right?
 - Do you want to look around for items?
 - There is a key on the floor, do you want to pick up the key?
 - There is a person blocking the door, do you want to speak to them?
 - The door is locked, do you want to check your bag for a key?
- The game takes the choices as input from the user.
- The user has to interact with the game and elements in the game until they reach an end point and complete the game.

You will need to design the structure of your game including where and how the players can move. You can create a text-based game or use a library to include graphics.

You will need to take a modular approach to the program to allow the user to move between different parts of the game, including back to previously visited positions. A class needs to be used for at least one element of the game.

The items that can be collected need to be carefully considered to make sure they can be identified by the player, picked up and stored in the bag. The bag will need to be an appropriate data structure to store a maximum of FOUR (4) items at a time.

The game needs to have a minimum of 10 different positions that the player can move to and a minimum of 5 different items to collect.

Task 1 - Decomposition and design – 40 marks

- Restate the specification and describe the outcomes.
- Identify a list of success criteria for the game to which you will compare your finished program. **(5 Marks)**
- Design the story for your game including a layout for the player to move in, the directions they can move at each stage, the items they can pick up and where these are used.
- Identify the data structures that will be required.
- Decompose the game into subproblems. **(5 Marks)**
- Create one or more structure diagrams for the game. **(5 Marks)**
- Design algorithms for all parts of the program using pseudocode. Design each of the sub-programs, for example functions, that will be used and how these will inter-relate. **(25 Marks)**

Task 2 Implementation – 30 marks

- Create your program using Python.
- Test your program regularly to make sure it works.
- Create a copy of your program code and include screenshots to show the implementation of your system.

Task 3 Testing – 25 marks

- Creating a testing strategy for your program to include:
 - Tests for all aspects of the program.
 - A range of test data include normal, extreme (where applicable) and not valid.
 - How the program has been put through unit testing and integration testing.
 - How the program will be tested as a whole using run-throughs and whitebox testing. **(15 Marks)**
- Test your program using your strategy and produce evidence of the outcomes using a test log.
- Complete the results of each test.
- Include evidence for each test. **(5 Marks)**
- Test your program against your success criteria. Produce evidence or refer to where the evidence can be found in your documentation, for the results of your testing. **(5 Marks)**

Task 4 Technical documentation – 5 marks

- Create an installation guide for your game to show a user how to install the program, where any files need to be located and how to start the program.
- Create technical documentation for your game to explain how your game works, the libraries it uses and the purpose of each function to allow other programmers to edit the game in future. This should include:
 - Commented code listings.
 - Identification and purpose of sub-programs.
 - Identification of libraries required, the library functions used and their purpose in the code.

Mark distribution

- Design (**40 marks**)
- Implementation (**30 marks**)
- Testing (**25 marks**)
- Publishing programme and installation documentation (**5 marks**)

Hints on design and implementation

- Your program does not need to make use of graphics, it can provide a text-output of game.
- Object-oriented programming must be used for one or more parts of the game.

Guidance

The assessment of your project will depend in part upon the quality of the documentation that you have produced.

- Restate the specification of the assignment by listing the Required Outcomes.
- Take the time required to design the assignment before you type any program code. Follow an appropriate design and documentation sequence.
- Always document your designs **before** you implement them.
- Provide a detailed design including, where appropriate, the design of any algorithms.
- Build in error handling to involve meaningful messages that would help with any future maintenance of the software.
- Annotate your program code appropriately.
- Design a testing strategy.
- Justify the design of suitable comprehensive test data.
- Show evidence of testing.
- Where appropriate, detail any major corrective action that you have taken in the light of the testing process.

Submission requirements

A word-processed document must be submitted incorporating the full documentation of all the significant aspects of the development of the assignment above. The document should be submitted both in paper form and digital form.

Refer to the Guidance above when producing your final documentation.

You are required to submit a publishable copy of the compiled system together with installation notes.

This publishable copy, that includes a setup file, should be on an appropriate medium (CD, USB flash drive, etc.).

Candidate checklist

Please use the following checklist to ensure that your work is ready for submission.

Have you read the NCC Education document *Academic Misconduct Policy* and ensured that you have acknowledged all the sources that you have used in your work?

Have you completed the *Statement and Confirmation of Own Work* form and attached it to your assignment? **You must do this.**

Have you ensured that your work does not contain viruses and can be run directly?

Example test log

| Test number | Test purpose | Input data | Expected result | Actual result | Evidence |
|-------------|--------------|------------|-----------------|---------------|----------|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |