Internal sample assessment tasks

NCFE Level 4 Award in Programming
QN: 603/7501/2

Version 1.0 January 2022

Contents

Introduction 3
Supervision of learners 4
Supporting learners 4
Unit 01 The fundamentals of programming (H/618/7116) 5
Unit 02 Developing a program in Python (K/618/7117) 7
Unit 03 Developing an app in C# (M/618/7118) 9
Unit 04 Using PHP for web development (T/618/7119) 11
Unit 05 Developing a web application using Java (A/650/0655) 12
Contact us 14

Introduction

We have created a set of sample tasks that you can contextualise to suit the needs of your learners to help them build up their portfolio of evidence. The tasks have been designed to cover all the learning outcomes for the unit and provide opportunities for stretch and challenge.

If you choose to create your own internal assessment tasks, they must:

- be accessible and lead to objective assessment judgements
- permit and encourage authentic activities where the learner's own work can be clearly judged
- permit effective discrimination between learners operating at different levels

More information can be found in the following document on our secure website delivering our qualifications – assessment and internal quality assurance guidance.

The learning outcomes for the units can be found in section 2 of the qualification specification.

Supervision of learners

Tutors should offer support and guide learners when carrying out work that is internally assessed.

Tutors should supervise the work carried out by learners to:

- monitor their progress
- prevent plagiarism
- ensure that any practical work is undertaken safely and in accordance with health and safety requirements
- ensure that the work completed is in accordance with the qualification specification and is suitable for internal assessment

Any work submitted for assessment must be authenticated and attributable to the learner.

The tutor must be satisfied that the work produced is the learner's own and the learner must declare that the work is their own.

Supporting learners

Tutors/assessors are also responsible for supporting learners through the assessment process to ensure that they are able to create and redraft/revise work independently.

Tutors/assessors may:

- help the learner to understand the concept of work-related work, applied learning and vocational qualifications
- help the learner to understand how to prepare and present their evidence, including what constitutes plagiarism and other forms of cheating
- motivate the learner to work consistently through the programme, including helping them work to deadlines
- encourage the learner to take the initiative in making improvements to their work but stop short of telling them the detail of the improvements to make
- provide reference material; however, model or worked answers should not be copied by the learner

Unit 01 The fundamentals of programming (H/618/7116)

Assessment task – unit 01

Task 1 (based on learning outcome and assessment criteria 4.2)

Create a report aimed at novice programmers, on how to identify and resolve syntax errors in programming code.

Task breakdown

The report should explain syntax errors, and it should show some real-life examples of how to identify and resolve syntax errors. Some examples of errors to include are:

- missing comma
- missing quotation mark
- missing brackets
- missing semicolon
- misuse of keywords
- undeclared variable
- misspelled variable
- missing operators
- misspelled instruction
- functions not called
- functions with no return value
- wrong number of arguments/wrong values passed to functions

The report should include suitable titles, screenshots and text.

Types of evidence

Evidence should include:

- a Microsoft Word/text document
- screenshots to demonstrate syntax errors

Task 2 (based on learning outcome and assessment criteria 5.1)

Create a presentation aimed at other programmers, providing a detailed account of emerging technologies in programming.

Task breakdown

The presentation should include contrasting perspectives on emerging technologies. The content could include:

- threading
- quantum programming
- machine learning

Assessment task – unit 01
The presentation should include suitable titles, images and text. The presentation should consist of a notes section: the software can be used to expand on some of the content covered in the presentation, which could include real-world examples, or an alternative Word/text document could be used. **Types of evidence** Evidence should include: • PowerPoint presentation, Prezi presentation • supplementary evidence in the form of a Microsoft Word/text document

Unit 02 Developing a program in Python (K/618/7117)

Assessment task – unit 02

Task 1 (based on learning outcome 2.2)

Design a flowchart to plan the code for the following program:

- create a quiz in Python, which asks various questions on a topic of your choice

Task breakdown

As a minimum, the program should:

- include questions that should increase with difficulty as they progress
- include comments to explain how each line of the program works
- give regular feedback throughout the quiz
- include a function to report a score at the end of each question and at the end of the quiz
- give a random piece of feedback based on the score at the end of the game

You must break down the functional requirements of the brief.

You need to design a flowchart, which should include:

- start/stop symbols
- process symbols
- decisions
- inputs/outputs
- direction/flow symbols

Once the flowchart is complete, produce the pseudocode to accompany the flowchart.

Types of evidence:

Evidence to be presented using relevant software packages. Evidence should include:

- screenshots (for example, of the flowchart)
- the pseudocode

Task 2 (based on learning outcome 3.1)

Create a quiz in Python, which asks various questions to students on a topic of your choice.

Task breakdown

As a minimum, your program should:

- include questions that should increase with difficulty as they progress
- include comments to explain how each line of the program works
- give regular feedback throughout the quiz

Assessment task – unit 02

- include a function to report a score at the end of each question and at the end of the quiz
- give a random piece of feedback based on the score at the end of the game

The program should consider the following:

- suitable data types
- variables
- appropriate operators
- use of sequence, selection and iteration
- use of lists
- use of an imported library
- proper use of Python syntax

You must run your code to ensure the program works correctly.

Types of evidence:

Evidence to be presented using relevant software packages. Evidence should include:

- screenshots (for example of the program/code)
- Python program file

Unit 03 Developing an app in C# (M/618/7118)

Assessment task – unit 03

Task 1 (based on learning outcome and assessment criteria 1.3)

Identify a development platform and based on this create a flowchart that documents the mobile application lifecycle, documenting the key events.

Task breakdown

Identify a development platform and justify why this is the best option for the task. If there are any constraints these must be highlighted.

Create a flowchart that shows the key mobile app lifecycle events and the order in which they can be expected to appear (for example, background then resume).

Note: The diagram should document common real-world application scenarios that would trigger a lifecycle event (for example, navigation, incoming call) and explain what common tasks a developer should complete when a lifecycle event is initiated (for example, save data).

Types of evidence:

Evidence should include:

- a document justifying choice of development platform and constraints
- a visual diagram using the tools of the learner's choice
- referencing to the stages within the diagram as required

Task 2 (based on learning outcome and assessment criteria 3.1)

Identify a development platform and based on this create a simple mobile application that presents a list of contact records and allows for the creation and storage of a new record.

Task breakdown

The presented application codebase should implement, as a minimum, the following core C# function areas:

- should present a view, bound to a viewmodel, that lists all contact records from a local database
- the contact list view should have an 'Add Button', which on pressing navigates to a new view that requests the contact's first name, surname and date of birth
- submission of the add contact view should be bound to an add record method in the viewmodel that saves the record to the local database

Where appropriate, the codebase could:

- validate that the first name and surname are complete and valid before submission
- create an edit view and associated viewmodel methods to allow for the editing of existing records

Assessment task – unit 03
Types of evidence: Evidence should include: • a working Visual Studio solution OR screenshots of the evidence

Unit 04 Using PHP for web development (T/618/7119)

Assessment task – unit 04

Task 1 (based on learning outcome and assessment criteria 1.3)

Create a simple set of PHP scripts that allow the user to enter their first name, surname and date of birth within a form that can be stored and retrieved from a PHP session.

Task breakdown

The presented scripts should, at the core, implement the following processes using PHP syntax:

- the scripts should collect user data (first name, surname, date of birth) from within an HTML form
- on submission of the form the backend script should be able to retrieve the submitted data
- the backend script should initiate a new session and store within the retrieved data
- the end user should be presented a link to a new PHP page to navigate to within their browser
- the new page should retrieve from the session, and present the originally submitted data
- the new page should destroy the session after output and demonstrate this

All code should be fully commented throughout.

Types of evidence:

Evidence should include:

- a set of working and commented PHP scripts (plain text formatted documents)

Task 2 (based on learning outcome and assessment criteria 2.2)

Create a technical project plan to implement a web-based contact management solution that involves the collecting, storing, and retrieval of contact address records.

Task breakdown

The plan should document the key functional areas of the system, such as:

- data entry (forms)
- data validation
- data storage (sessions and/or databases)
- data retrieval and display

The plan should expand in detail the data fields to be collected, the expected data types (for example, numbers, strings), the data that is required (validation) and how best to store and retrieve the data.

Types of evidence:

Evidence should include:

- a written plan (such as Microsoft Word/text document)

Unit 05 Developing a web application using Java (A/650/0655)

Assessment task – unit 05

Task 1 (based on learning outcome and assessment criteria 1.1)

Part A

Following the installation of a Java integrated development environment (IDE), explain the installation process and any various parameters entered. You will need to additionally provide an overview of the IDE and the steps involved installing a Java plugin.

Requirements

Create an overview document with embedded evidence and commentary covering:

- the installation process of a Java IDE
- a discussion of the system requirements explaining why these are important
- an overview of the IDE user interface capturing details of its structure and explaining its composite parts
- an explanation of what a Java plugin is and details of how to install a Java plugin in your IDE
- an explanation of the concept of the IDE workbench and a discussion of its composite parts

Part B

Using the installed IDE, create a simple ‘Hello World’ Java application.

Requirements

Your application should:

- use a relevant project name
- store the project files on the selected local disk
- capture details of the environment including screenshots where relevant
- include a screenshot of the completed project executing successfully

Types of evidence

Evidence should include:

- written report
- written answers
- screenshots with annotation
- program code listing
- observations

Assessment task – unit 05

Task 2 (based on learning outcome and assessment criteria 2.3)

Following the creation of a Java programme, it is important to test the application to ensure it is free from errors. Selecting the right testing method is an essential part of the testing.

Part A

Identify an appropriate testing method for your project and explain why you have chosen this over other methods.

Requirements

Create a document/presentation that considers common testing methods and identifies the best option to test your application.

You document should include:

- a range of common testing methods
- an overview of the method you intend to use
- justification for your choice

Part B

Create a test plan for your application that accurately checks the following:

- the application accepts the expected input
- the data is processed appropriately
- the output data is accurate and as expected
- the application functions as expected

Types of evidence

Evidence should include:

- written report
- observations
- presentation
- test plan

Contact us

NCFE
Q6
Quorum Park
Benton Lane
Newcastle upon Tyne
NE12 8BT

Tel: 0191 239 8000*
Fax: 0191 239 8001
Email: customersupport@ncfe.org.uk
Websites: www.ncfe.org.uk

DRAFT/Version 1.0 January 2022
Information in this internal assessment document is correct at the time of publishing but may be subject to change.

**** To continue to improve our levels of customer service, telephone calls may be recorded for training and quality purposes.***