HyperionDev

Curriculum title	Python Programmer
Curriculum code	900221-000-00-00
Module code	[module code]
NQF level	4
Credit(s)	40
Quality assurance functionary	QCTO - Quality Council for Trades and Occupations
Originator	MICT SETA
Qualification type	Skills Programme

Python Programmer

Learner Practical Skills Workbook

Name	
Contact Address	
Telephone (H)	
Telephone (W)	
Cellular	
Email	

Table of contents

Table of contents **2**
Practical Skills Module Specifications **4**
Provider Programme Accreditation Criteria **5**
Physical Requirements: 5
Human Resource Requirements: 5
Legal Requirements: 5
Exemptions: 5
Purpose of Practical Skills: Module 1 **6**
Practical Skills Module 1: Topic 1 7
Practical task **8**
Practical Skills Module 1: Topic 2 10
Practical task **11**
Practical Skills Module 1: Topic 3 12
Practical task **13**
Practical Skills Module 1: Topic 4 15
Practical task **15**
Practical Skills Module 1: Topic 5 17
Practical task **18**
Practical Skills Module 1: Topic 6 22
Practical task **23**
Share your thoughts **26**

Practical Skills Module Specifications

List of Practical Skill Module Specifications

900221-000-00-PM-01	Install computer software and hardware	L4	Cr3

You will demonstrate your competence in the knowledge topics and achievement of the assessment criteria through a process of continuous assessment (evaluation of your progress throughout the skills programme). These assessments involve interpreting evidence of your ability to perform specific tasks..

The practical tasks in this workbook must be submitted to the facilitator when you have completed them. They will be added to your portfolio of evidence (PoE), which will be signed by your facilitator as evidence that you have successfully performed these tasks.

Pay close attention to your facilitator's instructions and ensure you complete the activities within the given time.

Take note

This module has a total of 155 marks available. To meet the passing requirements, you will need to achieve a minimum of 124 marks, which represents 75% of the total marks. Achieving this threshold will ensure that you have met the necessary standards for this module.

Provider Programmme Accreditation Criteria

Physical Requirements:

- Valid licensed software and application, including OS
- Internet connection and hardware availability
- Examples and information specified in the scope statement and all the case studies, scenarios and access to hardware and software implied in the scope statements of the modules
- Remote learners: Provider must provide business IT simulation system (e.g. invoice processing)

Human Resource Requirements:

- Qualification of lecturer (SME):
    - NQF 5 industry recognised qualification with 1 year relevant experience
- Assessors and moderators: accredited by the MICT SETA

Legal Requirements:

- Legal (product) licences to use the software for learning and training
- OHS compliance certificate
- Ethical clearance (where necessary)

Exemptions:

- None, but the module can be achieved through RPL

Purpose of Practical Skills: Module 1

Programming Basics for Beginners, NQF 4

The focus of the learning in this module is on providing the learner with an opportunity to apply the basics of Python, IDE and Git commands and coding.

The learner will be required to:

- PM-01-PS01: Basic computer skills
- PM-01-PS02: Install Python
- PM-01-PS03: Getting started with an IDE
- PM-01-PS04: Getting started with GIT
- PM-01-PS05: Apply the programming life cycle to develop a solution
- PM-01-PS06: Create Python variables

Practical Skills Module 1: Topic 1

Topic Code	PM-01-PS01:
Topic	Basic computer skills

Scope of Practical Skill

Given an applicable instruction and access to a learning platform, the learner must be able to:

- PA0101 Operate computers using a range of functionalities:
    - Switch on, save, print, find a file, basic computer operating processes, task manager, operating systems etc.
    - Connect to the internet
    - Connect to a website
    - Use a search engine
    - Find and use cloud storage

Applied Knowledge

- AK0101 Concept, definition and functions of basic Python functionalities

Internal Assessment Criteria

- IAC0101 A range of basic computer functionalities are used successfully

Resources:

Knowledge	Information from Python Programmer Curriculum
National Curriculum Framework	900221-000-00-PM-01

Practical task

Follow the facilitator's instructions to complete the following activities:

Note:
- Create a folder with your name inside the task folder (e.g., 'Tim Jones').
- Once you have completed the practical module, ensure that this folder remains within the task folder so the facilitator can access your work.

Step 1: Switch on a Computer

Step 2: Save

Copy the following text into a Word document and save it as **Exercise1_Save** in your folder.
"An interesting fact about Python programming is that its name wasn't inspired by the snake—it actually comes from the British comedy series *Monty Python's Flying Circus*. Guido van Rossum, Python's creator, wanted a name that was short, unique, and a bit fun, reflecting the playful and approachable nature of the language! This is also why Python documentation and tutorials often include quirky examples and references to *Monty Python*."

Step 3: Open the "Exercise1_Save" document from your folder in Microsoft Word. Save it as a PDF by selecting "File" > "Save As," choosing "PDF" as the file type, and naming it "Exercise1_Save_PDF." Save this PDF in your folder.

Step 4: Choose any document on your computer (e.g., a Word file, PDF, or text file) or create a new one. Open the selected or created document, take a screenshot of its contents, and save the screenshot as "ChosenFile_Screenshot" in your folder. If you created a new document, also save the document itself in your folder.

Step 5: Access task manager. Save a screenshot “Task Manager” to your folder.

Step 6: Connect to the Internet using Microsoft Edge  or any browser of your choice.

Step 7: Connect to Python’s website using the below link: https://www.python.org/

Save a screenshot in your folder as “Python_Website”.

Step 8: Use the search engine “Google” (www.google.com) to search for facts about Python Programmers. Save a screenshot called “Python_SearchEngine”.

Step 9: Find and use cloud storage by accessing Google Drive. Save your screenshot “Python_SearchEngine” on Google Drive and take a screenshot. Save your screenshot as “GoogleDrive” in your folder on my Documents.

Take note

Important: Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.

Your Facilitator will complete the following evaluation checklist:

Check that the following is accomplished:

Item	Checked (Yes=5 No=0)	Comment: where did you find the evidence?
Switch on, save, print, find a file, basic computer operating processes, task manager, operating systems etc.		
Connect to the internet		
Connect to a website		
Use a search engine		
Find and use cloud storage		
Name of facilitator		
Signature		

Date	
Total	**/25**

Practical Skills Module 1: Topic 2

Topic Code	PM-01-PS02:
Topic	Install Python

Scope of Practical Skill

Given an applicable instruction and access to a learning platform, the learner must be able to:

- PA0201 Download and install Python
- PA0202 Troubleshooting/Verifying Python Installation
- PA0203 Download and install IDE to run Python

Applied Knowledge

- AK0201 Concept, definition and functions of basic Python functionalities

Internal Assessment Criteria

- IAC0201 Python is downloaded and installed
- IAC0202 Visual Studio Code is downloaded, installed and up and running with installation problems solved

Resources:

Knowledge	Information from Python Programmer Curriculum
National Curriculum Framework	900221-000-00-PM-01

Practical task

Follow the facilitator's instructions to complete the following activities:

Step 1: Download and install Python by following the instructions on the below link:

Download Python

Take screenshots during the Python download and installation process: one showing the download step (e.g., the website or file downloading) and one showing the installation step (e.g., the installer window or completion screen). Save them as “PythonDownload” and “PythonInstall” in your assignment folder.

Step 2: Troubleshooting Python Installation or Verify Python Installation

After installing Python, confirm that it works correctly before proceeding. Save a screenshot of the version output or any troubleshooting steps as “PythonVerify” in your folder.

Step 3: Download and install IDE to run Python. For purposes of this module it is

advised to download Visual Studio Code as an IDE. Download from the following link:

Visual Studio Code

After installation, open VS Code and take a screenshot of the interface (e.g., the welcome screen or an empty workspace) to confirm it’s ready for use. Save the screenshot as “VSCodeFinal” in your assignment folder.

Take note

Important: Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.

Your facilitator will complete the following evaluation checklist:

Check that the following is accomplished:

Item	Checked (Yes=5 No=0)	Comment: where did you find the evidence?
PA0201 Download and install Python		
PA0202 Troubleshooting Python Installation		
PA0203 Download and install IDE to run Python		
Name of facilitator		
Signature		
Date		
Total		**/15**

Practical Skills Module 1: Topic 3

Topic Code	PM-01-PS03:
Topic	Getting started with an IDE

Scope of Practical Skill

Given an applicable instruction and access to a learning platform, the learner must be able to:

- PA0301 Install IDE
- PA0302 Create a new Python project with IDE
- PA0303 Hello world: create a first Python programme
- PA0304 Save actions
- PA0305 Make the code more efficient and maintainable with refactoring with IDE

- PA0306 Perform debugging with IDE

Applied Knowledge

- AK0301 Concept, definition and functions of basic Python functionalities

Internal Assessment Criteria

- IAC0301 IDE is installed and successfully used for creating a new Python project
- IAC0302 “Hello world” is created

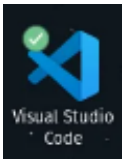
Resources:

Knowledge	Information from Python Programmer Curriculum
National Curriculum Framework	900221-000-00-PM-01

Practical task

Follow the facilitator's instructions to complete the following activities:

Step 1: You have already installed an IDE (Visual Studio Code) in Step 3 of PM-01-PS-02.

Choose the 

desktop shortcut to open the IDE. If no shortcut exists, search for “Visual Studio Code” in the Start menu and launch it.

Step 2: Create a new Python project with IDE (Visual Studio Code). Save this project as “Python Project”. Take a screenshot of VS Code showing the opened folder in the Explorer pane and save it as “PythonProject_Setup” in your folder.

Step 3: Hello World - Instructions

- Write a Python program that displays "Hello, World!"

- Save the program as a .py file (e.g., hello_world.py) in your task folder. Ensure the folder contains the .py file.
- In VS Code, click the “Run” button (triangle icon) in the top-right corner or select “Run” > “Run Without Debugging” to execute “hello_world.py.” You should see “Hello, World!” in the terminal below.

Step 4: Add a small enhancement to “hello_world.py” to make it more interesting, such as a variable.

Step 5: Check that your code adheres to **PEP 8 style guidelines**.

Take note

Important: Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.

Your facilitator will complete the following evaluation checklist:

Check that the following is accomplished:

Item	Checked (Yes=5 No=0)	Comment: where did you find the evidence?
PA0301 Install IDE		
PA0302 Create a new Python project with IDE		
PA0303 Hello world: create a first Python programme		
PA0304 Save actions		
PA0305 Make the code more efficient and maintainable with refactoring with IDE		
PA0306 Perform debugging with IDE		

Name of facilitator	
Signature	
Date	
Total	**/30**

Practical Skills Module 1: Topic 4

Topic Code	PM-01-PS04:
Topic	Getting started with GIT

Scope of Practical Skill

Given an applicable instruction and access to a learning platform, the learner must be able to:

- PA0401 Create and use a repository
- PA0402 Start and manage a new branch
- PA0403 Make changes to a file and push them to GitHub as commits
- PA0404 Open and merge a pull request

Applied Knowledge

- AK0401 Concept, definition and functions of basic Python functionalities

Internal Assessment Criteria

- IAC0401 Git functionalities are correctly used

Resources:

Knowledge	Information from Python Programmer Curriculum
National Curriculum Framework	900221-000-00-PM-01

Practical task

Follow the facilitator's instructions to complete the following activities:

Before starting this task, revisit the section on Git and GitHub to refresh your understanding of the key concepts. For additional guidance, refer to external resources such as **How to Create a Git Repository** and **Get started with GitHub documentation**

Step 1: Create and use a repository. Name this repository “Python_Repos”.

- Take a screenshot showing the repository creation result and save it in your folder. Name the screenshot: **repository_creation**.

Step 2: Create a new branch

- Create and switch to a new branch.
- Take a screenshot showing the branch created and active, then save it in your folder. Name the screenshot: **branch_creation**

Step 3: Make changes to a file and push them to GitHub as commits.

- Create or modify a file in your repository, such as README.md or a new file like **example.py.**
- Commit the changes and push them to Github
- Take a screenshot showing the updated file and commit reflected on GitHub, then save it in your folder. Name the screenshot: **commit_and_push**

Step 4: Open and merge a pull request.

- Open a pull request on GitHub to merge the changes from your new branch into the main branch, and complete the merge.
- Take a screenshot showing the successful pull request merge and save it in your folder. Name the screenshot: **pull_request**

Take note

Important: Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.

Your facilitator will complete the following evaluation checklist:

Check that the following is accomplished:

Item	Checked (Yes=5 No=0)	Comment: where did you find the evidence?
PA0401 Create and use a repository		
PA0402 Start and manage a new branch		
PA0403 Make changes to a file and push them to GitHub as commits		
PA0404 Open and merge a pull request		
Name of facilitator		
Signature		
Date		
	Total	**/20**

Practical Skills Module 1: Topic 5

Topic Code	PM-01-PS05:
Topic	Apply the programming life cycle to develop a solution

Scope of Practical Skill

Given an applicable instruction and access to a learning platform, the learner must be able to:

- PA0501 Apply the various stages (sequential and reiteratively) in the programming life cycle when developing an appropriate solution

Applied Knowledge

- AK0501 Concept, definition and functions of basic Python functionalities

Internal Assessment Criteria

- IAC0501 Expected results with the programming life cycle in Python are achieved

Resources:

Knowledge	Information from Python Programmer Curriculum
National Curriculum Framework	900221-000-00-PM-01

Practical task

Follow the facilitator's instructions to complete the following activities:

Save the steps followed as a screenshot “LifeCycle”.

Use the following scenario to complete the activities as set out:

Scenario: Developing a Weather Forecasting Mobile App

A team of developers is tasked with creating a mobile app that provides real-time weather forecasts. To ensure the app is built effectively, they apply the various stages of the programming life cycle, both sequentially and reiteratively, to develop the solution. Here's how each stage is applied:

---

1. Problem Definition

Sequential Application:
The team works with stakeholders (e.g., meteorologists and app users) to define the app's purpose and features:

The app should display current weather conditions, hourly forecasts, and a 7-day outlook.

Users should be able to search for locations and set preferences for units (e.g., Celsius or Fahrenheit).

Alerts for severe weather must be included.

Reiterative Application:
Feedback is collected from stakeholders, requiring the team to revisit and refine the problem statement, such as adding accessibility features like voice output for visually impaired users.

---

2. Feasibility Study and Planning

Sequential Application:
The team analyses whether the app is feasible within budget, timeline, and resources. They outline:

The technical feasibility of integrating APIs for weather data.

The cost of development tools and hosting services.

A timeline for each development phase.

Reiterative Application:
Unexpected challenges (e.g., the chosen API has limited accuracy) prompt the team to revise their plans and find a better API provider.

---

3. System Design

Sequential Application:
The team designs the app's architecture, including:

Frontend: User interface with intuitive navigation and a visually appealing design.

Backend: API integration for real-time weather data retrieval.

Database: Storing user preferences and location data.

Reiterative Application:
During testing, users find the interface confusing, requiring the team to redesign the navigation flow for better usability.

---

4. Implementation

Sequential Application:
Developers write the code based on the system design.

The backend team integrates weather APIs.

The frontend team develops the user interface.

Reiterative Application:
As bugs are identified during development, developers revisit and rewrite parts of the code to optimize performance and ensure compatibility across devices.

---

5. Testing

Sequential Application:
The app undergoes unit testing, integration testing, and system testing to ensure all components work together.

Unit Testing: Checks individual modules like location search functionality.

Integration Testing: Verifies that the weather API delivers accurate data to the frontend.

System Testing: Confirms the app functions as expected on different devices.

Reiterative Application:
Bugs or errors identified during testing are fixed, and the app is tested again. For example, incorrect time zone data requires changes in the API handling code.

---

6. Deployment

Sequential Application:
Once testing is complete, the app is released on app stores (Google Play and Apple App Store).

Reiterative Application:
After deployment, user feedback highlights issues, such as slow loading times, requiring developers to revisit the code and release updates.

---

7. Maintenance

Sequential Application:
The app is monitored for performance and updated regularly to ensure it remains functional with new devices and operating systems.

Reiterative Application:
New features (e.g., adding radar maps or air quality indicators) are developed and integrated into the app based on user suggestions, restarting the development life cycle.

By applying the programming life cycle both sequentially and reiteratively, the development team ensures the weather app meets user needs, adapts to changing requirements, and evolves over time to remain competitive.

Apply the stages of the programming life cycle to the given Weather Forecasting App scenario. For each of the 7 stages, you must describe:

- How that stage would be applied in sequence (done step-by-step).
- How that stage would be applied iteratively (done again based on feedback, errors, or new requirements).

These stages are:

1. Problem Definition
2. Feasibility Study and Planning
3. System Design
4. Implementation
5. Testing
6. Deployment
7. Maintenance

Complete the activity in a document (e.g., Google Docs or Microsoft Word) and save the document with the filename **LifeCycle** and convert it to a PDF before submission

Take note

Important: Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.

Your facilitator will complete the following evaluation checklist:

Check that the following is accomplished:

Item	**Checked (Yes=5 No=0)**	**Comment: where did you find the evidence?**
PA0501 Apply the various stages (sequential and reiteratively) in the programming life cycle when developing an appropriate solution		
Name of facilitator		
Signature		
Date		
	Total	**/5**

Practical Skills Module 1: Topic 6

Topic Code	PM-01-PS06:
Topic	Create Python variables

Scope of Practical Skill

Given an applicable instruction and access to a learning platform, the learner must be able to:

- PA0601 Use keywords
- PA0602 Print a simple string
- PA0603 Print blank lines
- PA0604 Print end command
- PA0605 Create and assign value
- PA0606 Assign multiple values to multiple variables
- PA0607 Declare and assign value to constant.
- PA0608 Assign integer literals into different variables
- PA0609 Define and declare string variable types
- PA0610 Re-declare a variable
- PA0611 Delete a variable
- PA0612 Correct and debug errors

Applied Knowledge

- AK0601 Concept, definition and functions of basic Python functionalities

Internal Assessment Criteria

- IAC0601 Expected results with Python variables are achieved

Resources:

Knowledge	Information from Python Programmer Curriculum
National Curriculum Framework	900221-000-00-PM-01

Practical task

Follow the facilitator's instructions to complete the following activities:

Build a small program that manages student information and integrates the concepts of variables, keywords, constants, and debugging by following step 1 to step 12. Consolidate all tasks into a single Python file named **student_info_manager.py** and ensure it is uploaded to the relevant folder.

Step 1: Use keywords

Create a condition to check if a student passed or failed a test using the if and else keywords.

Requirement: A student passes if their grade is 50 or more.

Step 2: Print a simple string

Start your program with a message like “Welcome to the Student Information Manager!”.

Step 3: Print blank lines

Use print() with no arguments to separate different parts of your output for better readability.

Step 4: Print end parameter in Print

Show that you can print multiple pieces of information on the same line using print(..., end="").

Step 5: Create and assign value

Define variables to store student name and student grade.

Step 6: Assign multiple values to multiple variables

Use one line of code to assign values to three variables: name, grade, and age.

Step 7: Declare and assign value to constant.

Create a constant for the passing grade using all uppercase letters (e.g., PASS_MARK = 50).

Step 8: Assign integer literals into different variables

Create integer variables that represent values like student ID, enrolment year, or completed credits.

Step 9: Define and declare string variable types

Store and display the student's first name and last name using string variables.

Step 10: Re-declare a variable

Show that you can change the value of an existing variable — for example, update the student's grade after re-evaluation.

Step 11: Delete a variable

Use the del keyword to delete a variable (e.g. age), and comment on what happens if you try to access it afterward.

Step 12: Ensure Code Quality

Ensure that your code follows PEP 8 style guidelines for readability and consistency, and aim for efficient implementation of all tasks.

Take note

Important: Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.

Your facilitator will complete the following evaluation checklist:

Check that the following is accomplished:

Item	Checked (Yes=5 No=0)	Comment: where did you find the evidence?
PA0601 Use keywords		
PA0602 Print a simple string		
PA0603 Print blank lines		
PA0604 Print end command		
PA0605 Create and assign value		
PA0606 Assign multiple values to multiple variables		
PA0607 Declare and assign value to constant.		
PA0608 Assign integer literals into different variables		
PA0609 Define and declare string variable types		
PA0610 Re-declare a variable		
PA0611 Delete a variable		
PA0612 Correct and debug errors		

Name of facilitator	
Signature	
Date	
Total	**/60**

Requirements met **124+/155**		**Total**	**/155**
Requirements not met **Under 124/155**			
Facilitator signature		**Date**	

Share your thoughts

Please take some time to complete this short feedback **form** to help us ensure we provide you with the best possible learning experience.