Open Awards

Quality Endorsed Unit

1 Unit Details

Unit Title:	Python Programming
Unit Code:	CK3/3/WR/012
Level:	Level 3
Credit Value:	8
GLH:	56

2 Learning Outcomes and Criteria

Learning Outcome (The Learner will):	Assessment Criterion (The Learner can):
1. Understand how to install the Python interpreter and run Python programs	1.1 Complete a Python installation under the Microsoft Windows operating system
	1.2 Use IDLE to input Python commands and statements
	1.3 Create a simple Python program and run it in both IDLE and in a command prompt window
2. Understand Python variables and data types	2.1 Illustrate the use of variables, their naming conventions, and scope in a Python program
	2.2 Incorporate Python strings and a range of string methods into a Python program
	2.3 Perform basic arithmetic using a range of Python's arithmetic operators
	2.4 Deploy Python's comparison and logical operators in a Python program
	2.5 Manipulate and concatenate Python strings and work with a range of numeric data types
3. Understand Python's branching and looping statements	3.1 Incorporate the IF, ELSE, and ELIF statements into a Python program
	3.2 Insert FOR and WHILE loops into a Python

			program
		3.3	Create a “Guess the Number” game program combining Python’s branching and looping statements
4.	Understand Python Tuples, Lists, and Dictionaries	4.1	Create a “Guess the Word” game and: a) Store words in a tuple b) From a random word, replace alternative letter with a hyphen c) Ask the user to guess the word d) Check the guess is correct
		4.2	Create a Python List and: a) Add an item to the list b) Insert an item between existing items c) Delete a list item d) Add an additional list e) Organise the lists into ascending and descending orders f) Create nested lists
		4.3	Create a Python Dictionary to: a) Contain a range of key-value pairs b) Print using a FOR loop c) Add a new entry d) Modify an existing entry e) Delete an entry
5.	Understand Python functions	5.1	Create a Python function, passing arguments to the function and returning values from it
		5.2	Create an area calculation program to: a) Ask the user to enter width and height b) Calculate area function c) Return the calculated area to the main program
		5.3	Create a Python Hangman game to: a) Display the progress of the scaffold b) Include local variable c) Include global variables
6.	Understand how to create and access Python files	6.1	Create a Python file and: a) Add records to the file b) Amend existing records c) Close the file d) Use a range of file object methods
		6.2	Read both characters and lines from a Python file
		6.3	“Pickle” text data into a binary stream and “unpickle” the binary stream into plain text
		6.4	Create a shelf file containing a range of lists and: a) Close the file b) Open the file in Read Only mode c) Unpickle the data d) Print the lists in there original format
		6.5	Recognise the various exception types, trap errors using Try and Except, and raise exceptions
7.	Understand Python objects	7.1	Create a class with multiple methods and instantiate multiple objects from the class

		7.2	Create a constructor method with attributes
		7.3	Create a class with private attributes and methods, create multiple properties, and access the properties from outside of the class
8.	Understand Object-Oriented Programming techniques	8.1	Build a multi-class “Highest Card” game, which includes interacting Game, Deck, Hand, and Card classes, with multiple methods and functions
		8.2	Incorporate inheritance techniques into the “Highest Card” game, changing inherited methods, including polymorphic behaviour, and creating modules
9.	Understand Python Graphical User Interface (GUI) development and event-driven programming techniques	9.1	Create a GUI window that responds to mouse clicks and includes: a) Objects and widgets b) Text boxes c) Data entry boxes d) List boxes e) Check boxes f) Radio buttons g) Colours h) Images
		9.2	Use the Grid Layout Manager to accurately position widgets in a GUI window
10.	Know how to develop graphical applications using the Pygame and Livewires packages	10.1	Create a graphics window containing a background image and moving “sprites”, both automatically and via the mouse
		10.2	Incorporate collision detection and collision handling techniques into a moving sprites graphics window
		10.3	Develop a ping pong game where a “bouncing” ball has to be prevented from reaching the bottom of the graphics window by intercepting it with a paddle, controlled by the user’s mouse
		10.4	Develop game applications using both mouse and keyboard input and including sounds and animation techniques