**Assessment type (🗹):**

Questioning (Oral/Written)

Practical Demonstration

3rd Party Report

Other – Lab

**Assessment Resources:**

|  |
| --- |
| The base requirements this assessment task include:   * IDE or editor for developing Python programs (only IDLE and PyCharm supported by the college) * Access to Office 365 & Microsoft Word * Virtual machine   You may not need all these for every part in this assessment |

**Assessment Due :**

|  |
| --- |
| This assessment is due after the weekly session, **Week 9, Friday 17:00**. |

**Assessment Instructions:**

|  |
| --- |
| 1. Your code must be written in IDLE or PyCharm IDEs. If you are using a different IDEs or a different structure for your application, then assistance from your lecturers may be limited (at best).  Discuss with your lecturer before straying too far off the path! 2. All resources used should be referenced with the question. Answers may not be copied and pasted from any resource. All answers must be reworded to display your understanding. 3. You may only use Python functionality, methods and libraries which were taught in this unit. 4. First line of code in a program should have the student’s name and number, as proof of authenticity. 5. Screenshots of all programs must be included **in** this document, with the appropriate question. 6. Screenshots of testing, showing your code works as intended, should be included with the relevant question. 7. Python programs should be named: XXX\_Lab##\_SYY\_QZZ   Replace XXX with your initials Replace ## with Lab number Replace YY with Section number,  Replace ZZ with Question number   1. It is a submission requirement that all screen shots be signed in some way. Some acceptable examples of signed screen shots are shown below.  **Example 1**: Signed using a simple drawing tool.   **Example 2**: Water marked signature.  A screenshot of a computer     Description automatically generated   **Example 3**: Program named as prescribed, as well as first line comment with student name and number. Program saved as pre-described.   1. All python programs must be included in the submission, as well as this document. |

**Assessment Instrument:**

# Section 1: Defining dictionaries

|  |
| --- |
| Begin by answering the following questions about dictionaries. |
| 1. What is a dictionary? |
| In Python dictionaries are used to store data values in key:value pairs.  A dictionary is a collection which is ordered and changeable. |
| 1. What makes dictionaries different to lists? |
| Dictionaries doesn’t allow duplicate values and are written with curly brackets. |
| 1. Can dictionaries contain lists? |
| Yes |
| 1. What are two ways you can create a dictionary using Python? |
| By using the constructor thisdict = dict(name = "John", age = 36, country = "Norway")  Or, key: value pairs surrounded by curly brackets  thisdict = { name: "John", age: 36, country: "Norway"} |

# Section 2: Working with a simple dictionary

|  |
| --- |
| Complete the following tasks using the interpreter. |
| 1. Create a dictionary called **translations** and assign the following key/value pairs to it. 2. Print the complete dictionary. |
|  |

|  |
| --- |
| 1. Print the value for the key **two** from the dictionary using square brackets. |
|  |

|  |
| --- |
| 1. Print all items in dictionary as tuples, e.g. |
|  |

|  |
| --- |
| 1. Print only the **keys** in the dictionary |
|  |

|  |
| --- |
| 1. Print only the **values** in the dictionary. |
|  |

|  |
| --- |
| 1. Add the key value pairs **“twenty”:”zwanzig”** and **“twenty one”:”ienundzwanzig”** to the dictionary. 2. Remove the key pair **“zero”:”null”** from the key pair 3. Print the dictionary one last time. |
| **Output** |
|  |

# Section 3: Nested dictionaries

|  |  |
| --- | --- |
| Complex dictionaries can be made from any number of components (including other dictionaries), particularly when automating data input from one application to another. Complete the following tasks using a code file.  Open the code file “**games.py**” from BlackBoard and complete each of the challenges in the python file. Your goal is to demonstrate you can work with the nested dictionaries by extracting the required data and formatting it in the exact way presented by the demo screen shots.  *To be marked as satisfactory for this section, you must provide:*   * 1. Your own screen shots matching the examples.   2. Screenshots as well as your final code in a code file.   3. Your final JSON file. | |
| 1. Challenge 1: Print all game names using a loop. | |
| **Code** | **Output** |
|  |  |

|  |
| --- |
| 1. Print the name and cost of all games that cost less than 50 dollars. |
| **Code** |
|  |
| **Output** |
|  |

|  |
| --- |
| 1. Print the details of the game with the ID **312530**. |
| **Code** |
|  |
| **Output** |
|  |

|  |
| --- |
| 1. Find and display all the games with the tag “Action” using a nested loop. |
| **Code** |
|  |
| **Output** |
|  |

|  |
| --- |
| 1. Change the price of any game the user wishesto change to any value a user wants to input. Ensure you preserve data types and display the new game information to confirm. |
| **Code** |
|  |
| **Output** |
|  |

|  |  |
| --- | --- |
| 1. Add the following code to the file to save the information in a structured JSON file. 2. Run your complete program and verify the data was saved to the JSON file (relative pathing is used in the JSON file path, so it should save to the same location as the script you are working with). | |
| **Code** | **Output** |
|  |  |

# Section 4: JSON

|  |
| --- |
| Answer the following questions and complete the following tasks. Supporting documentation can be found [here](https://www.programiz.com/python-programming/json). |
| 1. What is **JSON**? |
| Mostly used for data transfer between web servers and web applications, JSON (JavaScript Object Notation) is a text-based, human-readable format for storing and sharing structured data. It can also be used to store unstructured data. |
| 1. What are the advantages to saving data in the JSON format? |
| JSON data storage has several benefits, including being self-describing and human-readable, which makes debugging and development easier, and being lightweight and space-saving, which makes it perfect for speedier data transmission. |
| 1. What is **YAML** and **XML**? |
| **YAML:** A human-readable data serialization language, YAML is meant to be simple to read and write.  **XML:** Like HTML, XML is a markup language that defines elements and their structure using tags. |

|  |  |
| --- | --- |
| 1. Create a new code file called **games2.py**. The code file should:    1. Open the JSON file saved from the last section.    2. Read all the data in JSON format.    3. Import the JSON data back into a dictionary exactly as it was saved.    4. Print the data back (no formatting required), just verification the data was read and is the correct type. | |
| **Code** | **Output** |
|  |  |