**Assessment type (🗹):**

Arvand Eftekhar 20108797

Questioning (Oral/Written)

Practical Demonstration

3rd Party Report

Other – Project/Portfolio (*please specify)*

**Assessment Resources:**

|  |
| --- |
| You do not need specific resources to answer the questions in this assessment.  You may, however, use a Python shell and the online Python documentation if you think that helps. |

**Assessment Instructions:**

|  |
| --- |
| This assessment consists of multiple questions. Answer all questions to the best of your ability. Think about your spelling and grammar. If a question has a minimum or maximum word count, take that into account.  Answer all questions in your own words. If you use external resources, including ChatGPT, please provide references.  If provided, you should use the provided template to answer your questions.  Do NOT zip the Word document before uploading it. |

**Assessment Instrument:**

|  |
| --- |
| Question 1 Fundamentals of OOP (100-200 words)  1. **In your own words** describe what an object is and its relationship to a class. 2. An object contains behaviours, state, and boundaries. How do specify each in Python? 3. What are two key problem that OOP is trying to solve? |
| An object is like a thing in the real world, like a car, a dog, or a person. It has characteristics and things it can do. A class is like a blueprint or a template for creating objects. It defines what the object will be like and what it can do.  1. In Python, you specify behaviours of an object by defining functions inside the class. These functions are called methods. The state of an object refers to its current condition or data. You specify the state by defining variables inside the class, which are called attributes. 2. **1-Complexity Management: OOP helps manage complexity by breaking down a program into smaller, more manageable parts called objects. Each object has its own data and behaviours, which makes it easier to understand and maintain the code.**   **2-Code Reusability: OOP promotes code reusability by allowing objects to be reused in different parts of a program or in different programs altogether. Once a class is defined, it can be used to create multiple objects with similar characteristics and behaviours, saving time and effort in writing new code.** |
| Question 2 – Language constructs (20-50 words) In your own words, describe what a sequence is. Illustrate with an example. |
| A sequence is an ordered collection of elements. In programming, it's often used to store multiple values in a specific order. For example, a list in Python is a sequence: my\_list = [5, 6, 7, 8, 9]  Here, **my\_list** is a sequence of numbers in the order they appear. |
| Question 3 – Language constructs (20-50 words)  1. In your own words, describe what iteration is. 2. Provide at least two iteration constructs that are supported in Python and provide an example of one. |
| a. Iteration is the process of repeatedly executing a set of instructions. It allows you to perform actions multiple times, typically over a sequence of elements.Two iteration constructs in Python are for loops and while loops. Here's an example of a for loop: # Iterating over a list  my\_list = [5, 6, 7, 8, 9]  for num in my\_list:  print(num)  This loop iterates over each element in **my\_list** and prints it. |
| Question 4 – Language constructs (20-50 words) In your own words, describe what a selection is. Illustrate with an example. |
| Selection, also known as conditional execution, involves making decisions based on certain conditions. It allows a program to choose between different paths of execution based on whether a condition is true or false.Example: # Checking if a number is positive or negative  num = 10  if num > 0:  print("The number is positive.")  else:  print("The number is negative.")  In this example, the program selects different paths based on whether the variable num is greater than zero or not. If the condition num > 0 is true, it executes the code inside the if block, otherwise, it executes the code inside the else block. |
| Question 5 – Language syntax rules (30-70 words)  1. In your own words, describe the importance of indentation in Python. 2. Provide at least one example of code where a difference in indentation changes the meaning of the program. |
|  |
| Question 6 – Data types (20-50 words)  1. In your own words, describe how a user might input a number (in a terminal, from the keyboard) and how you can use that number within a calculation. 2. You may illustrate with a small code snippet. Be precise and make sure Word’s auto correct does not change the meaning of the program. |
|  |
| Question 7 – Data types (20-50 words)  1. Describe the following Python data types and give an example of each: a) Integer b) Float c) String d) Boolean |
|  |
| Question 8 – OOP Principles (20-50 words)  1. Explain the concept of polymorphism and inheritance in object-oriented programming. Provide an example in Python to illustrate these concepts. |
|  |
| Question 9 – Organisational documentation (20-50 words) In your own words, describe the importance of PEP-8 and why organisations often use this as their preferred coding style for Python. |
|  |
| Question 10 – Constructors (30-70 words) Explain the purpose of constructors in Python classes. Provide an example of a constructor in a Python class. For the purposes of this question, the \_\_init\_\_ method can be considered a constructor. |
|  |
| Question 11 – Aggregation (30-70 words)  1. Describe the concept of object aggregation and provide a Python example to illustrate this. |
|  |
| End of the questions |