|  |  |  |  |
| --- | --- | --- | --- |
| Instructor |  | Due Date |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part** | **1** | **2** | **3** | **4** | **Total** |
| *Maximum Points* | **25** points | **25** points | **25** points | **25** points | **100**G101010 pointsG |
| ***Your Score*** |  |  |  |  |  |

**Textbook Reading Assignment**

Thoroughly read Chapter(s) on Dictionaries, Lists, Tuples and Arrays in your **Python** textbook.

**Part 1 Glossary Terms**

Define, in detail, each of these glossary terms from the realm of computer programming logic and design and computer topics, in general. If applicable, use examples to support your definitions. Consult your notes or course textbook(s) as references or the Internet by visiting Web sites such as:

|  |  |  |
| --- | --- | --- |
| http://www.askjeeves.com | http://www.webopedia.com | http://www.bing.com |

**(a) dictionary**

|  |
| --- |
| A Python type with key value pairs, e.g., {“key”:”value”}. Values are returned using bracket notation (my\_dict[“some\_key”]) or the my\_dict.get() method |

**(b) element ( of an array )**

|  |
| --- |
| Item in an array, could be any type, e.g, my\_list = [“some element”, “another element”, [“list”,”element”,”inside”,”list”]] etc. |

**(c) index**

|  |
| --- |
| The number that indicates the position of an element in a list, e.g., my\_list[0] returns the first item in a list |

**(d) list**

|  |
| --- |
| A Python type that is a collection of any data type |

**(e) tuple**

|  |
| --- |
| A Python type that contains comma separated values similar to a list, but is immutable (cannot be changed) and is often used for cases when developers want to protect certain data from corruption |

**Part 2 Textbook Exercises - Dictionaries, Lists, Tuples and Arrays**

For each of the following, select the correct answer.

**(1)** The indexes of an array always start with position number 0 .   
  
 **(a) True**  (b) False

**(2)** The last index of an array equals the length of the array less 1 .

(**a) True** (b) False

**(3)** Given the following code, what will total be equal to?

**from array import \***

**SIZE = 5**

**abc = array("i", [1, 2, 3, 4, 0 ])**

**total = 1**

**for index in range(2, SIZE) :**

**total = total + abc[index]**

**print(total)**

(a) 5 (b) 35 (c) 7 (d) 10 **(e) 8**

**(4)** What is the range of valid indices for the following?

**from array import \***

**myArray = array("i", [0, 1, 0, 1, 0, 1, 0, 1, 0, 1])**

(a) 0 - 10 (b) 1 - 10

(c) To infinity and beyond infinity. **(d) None of these**

**(5)** What is the range of valid indices for the following?

**from array import \***

**myList = [10 for i in range(101)]**

**(a) 0 - 100** (b) 1 - 101

(c) To infinity and beyond infinity. (d) 0 - 101

**(6)** The indexes of an array always start with 0 and end with the integer that is

**(a) one less than the size of the array.**

(b) one greater than the size of the array.

(c) the actual size of the array.

(d) twice the size of the actual array.

(e) None of the above

**(7)** What is the output of the following code?

**from array import \***

**a = array("i", [1, 2, 3])**

**print(a[2], " ", a[1], " ", a[0])**

(a) 1 2 3 (**b) 3 2 1** (c) 1 3 2 (d) 2 1 3 (e) 3 1 2

**(8)** What is the output of the following code?

**my\_list = [10, 20, "Cindy", 30, "Jill"]**

**my\_list.append(40)**

**print(my\_list[0] + my\_list[3] - my\_list[5])**

(a) Cindy (b) 10 (c) 1020 **(d) 0** (e) Jill

**(9)** How many dictionary items elements are there after this block of statements is executed?

**# telephone book ( extensions )**

**phonebook = {"James" : 858, \**

**"Katherine" : 837, "Luke" : 800, \**

**"Alexandria" : 802}**

**print(phonebook)**

**phonebook["Gregory"] = 807**

**print(phonebook)**

**del phonebook["Luke"]**

**if ("Luke" in phonebook) :**

**print("dictionary item found")**

**else :**

**print("dictionary item not found")**

**phonebook["Luke"] = 805**

**for key, value in sorted(phonebook.items()) :**

**print(key, "\t", value)**

(a) 1 (b) 0 **(c) 5** (d) 2 (e) 7

**(10)** After this block of statements is executed, which letters will remain in the list?

**# list of letters**

**my\_list = ["p", "y", "t", "h", "o", "n"]**

**my\_list.append("g")**

**my\_list.sort()**

**my\_list.pop()**

**my\_list.pop()**

**my\_list.remove("p")**

**my\_list.append("i")**

**my\_list.reverse()**

**print(my\_list[1 : 3])**

(a) h and i (b) i and n

**(c) o and n** (d) p and n

**Part 3 Textbook Exercises - Dictionaries, Lists Tuples and Arrays**

**(1) ( Single - Dimensional Arrays )**

Examine the script given below and then modify it such that the array will hold six integers. Then run the program and input six distinct integers.

Paste a snapshot of the output in the space provided.

|  |
| --- |
| **from array import \***  **SIZE = 4**  **originalArray = array('i', [0, 0, 0, 0])**  **reverseArray = array('i', [0, 0, 0, 0])**  **print ("enter the original array elements")**  **for index in range(0, SIZE) :**  **print(" please enter an integer: ")**  **originalArray[index] = int(input())**  **print ("\noriginal array")**  **for index in range(0, SIZE) :**  **print(originalArray[index])**  **print ("\narray reversed")**  **for index in range(SIZE - 1, 0 - 1, -1) :**  **reverseArray[index] = originalArray[index]**  **print(reverseArray[index])**  **originalArray.reverse()**  **print ("\nreverse array")**  **for index in range(0, SIZE) :**  **print(originalArray[index])** |

**(2) ( Single - Dimensional Arrays )**

Examine the script given below and then modify it by supplementing the code statements with a function that will determine the minimum value in the array.

Paste a snapshot of the output in the space provided.

|  |
| --- |
| Text  Description automatically generated |

**Part 4 Textbook Exercises - Dictionaries, Lists Tuples and Arrays**

**(1) ( Single - Dimensional Arrays )**

Write a complete program that declares an array of any five integers, from 0 to 100 , and averages only those integers which are greater than 60 .

*import* random

random\_arr = []

*for* i *in* range(5):

random\_arr.append(random.randrange(1,101))

print(random\_arr)