Started on	Wednesday, 6 December 2023, 12:35 PM
State	Finished
Completed on	Wednesday, 6 December 2023, 12:37 PM
Time taken	1 min 8 secs
Marks	13.00/13.00
Grade	**10.00** out of 10.00 (**100**%)

Question 1

Correct

Mark 1.00 out of 1.00

To find the address of any given element in an Array, in memory, we can use this equation:

indexAddress = baseAddress ✔ + (elementSize ✔ * index ✔)

indexAddress

elementSize

baseAddress

indexAddress

Your answer is correct.

Question 2

Correct

Mark 1.00 out of 1.00

To initialise an array, you **MUST** explicitly specify the size **OR** its contents must be explicitly defined.

- True ✔
- False

Question 3

Correct

Mark 1.00 out of 1.00

Consider the following line of C# code:

```
int[] myArray = new int[6];
```

What is the index of the last (highest) **index** of myArray?

- a. 5 ✔
- b. 6
- c. 7
- d. 0

Your answer is correct.

Question 4

Correct

Mark 1.00 out of 1.00

In the example shown below:

```
int[,] myArray = new int[20,15];
```

What type of array is being intialised?

- a. 1 Dimensional, Non-Jagged Array
- b. 2 Dimensional, Non-Jagged Array ✔
- c. 2 Dimensional, Jagged Array
- d. 1 Dimensional, Jagged Array

Your answer is correct.

Question 5

Correct

Mark 1.00 out of 1.00

To initialise a List, you **MUST** declare the data **type** being stored.

- True ✔
- False

Question 6

Correct

Mark 1.00 out of 1.00

What answer below best describes what happens to a List when a new item is appended, using the .Add() command?

- a. The list only ever contains enough space for the current number of items. Calling add results in a new list being created and all values being copied over. Then the new item is added to the new empty space.
- b. If there is space (capacity), the item is added to the end of the list. If there is not enough space the list is resized and the new item ✔ is added, size is incremented by one.
- c. As lists are always fixed-size and cannot change size after creation, the least recently accessed element is replaced with the new value.
- d. If there is space (capacity), the item is added to the end of the list. If there is not enough space the item is silently discarded.

Your answer is correct.

Question 7

Correct

Mark 1.00 out of 1.00

Strings, in C#, are **immutable**. This means it cannot be changed once it has been created.

- True ✔
- False

Question 8

Correct

Mark 1.00 out of 1.00

A String is an example of a collection.

What expression below **BEST** describes this collection?

- a. IEnummerable<char> ✔
- b. List<char>
- c. Dictionary<int, char>

Your answer is correct.

Question 9

Correct

Mark 1.00 out of 1.00

A Dictionary **DOES NOT** require the data **types** of the key and value, to be declared when the dictionary is created.

- True
- False ✔

Question 10

Correct

Mark 1.00 out of 1.00

BOTH the types of the **KEY** and **VALUE** of a Dictionary entry **MUST** be of the same data type

- True
- False ✔

Question 11

Correct

Mark 1.00 out of 1.00

To initialise a HashSet you **DO NOT** require the data type.

- True
- False ✔

Question 12

Correct

Mark 1.00 out of 1.00

In C#, how would we define a Set/HashSet?

Sets are [unordered] ✔ collections of [unique] ✔ elements.

[hierarchical] [untyped] [ordered]

[labelled] [duplicate] [paired]

Your answer is correct.

Question 13

Correct

Mark 1.00 out of 1.00

Given the following code:

```
var numbers = new HashSet<int>{1,6,3,7,8};
```

What happens if I run the operation below?

```
numbers.Add(6);
```

a. A runtime error/exception occurs.

b. This code would not compile - sets can only contain class types, and int is not a class type.

c. The HashSet adds 6 to the set.

d. Nothing happens. ✔

e. This code would not compile - sets cannot be modified after creation

Your answer is correct.

◄ Worksheet 4 Brief

Jump to...

Formative Queue for Data Structures ►