Bernardo Fitzmaurice Acevedo 105297603

Class code: COS30031

Start date: 02/08/2024

Task 02 Lab C++ for Programmers

* Q.1 What is the difference between a struct and a class?
  + Structs is a group of values, while class is a type with a value defined by the user.
* Q.2 What are function declarations?
  + Bind a function to a name, and I give it a return type and parameters.
* Q2.1 Why and when are they needed?
  + They are needed so that the function can be used throughout the code, and they are used when you create a function that you want to use in more than one part of the code.
* Q.3 Why are variable names not needed here?
  + Because the only the function’s interface is specified.
* Q3.1 Could you add variable names? Would that be good?
  + Yes, I could add more variables. I don’t think it would be good, because it’s not necessary.
* Q.4 Does your IDE know if this method is used? If yes - how does it indicate this? (Colour? Tip? Other?)
  + Yes, it can know if it’s used. It grays out the color of the function if it’s not used.

**Section 1**

* Q.5 un-initialised values ... what prints and why?
* Q.6 Did this work as expected?
* Q.7 Initialisation list - what are they?
* Q.7.1 Does your IDE suggest what the values are?

**Section 2**

* Q.8 Should show age=1, x=1, y=2. Does it?
  + No, age is 1, but x is 2 and y is 3.
* Q.9 Something odd here. What and why?
  + The age is odd, it marks age as 4294967295.

**Section 3**

* Q.10 showParticle(p1) doesn't show 5,6,7 ... Why?
  + Because p1 is set as {1,1,1} **(Check answer again)**

**Section 4**

* Q.11 What does -> mean?
  + Is an access operator.
* Q.12 Do we need to put ( ) around \*p1\_ptr?
* Q.13 What is a dereferenced pointer?
* Q.14 Is p1 stored on the heap or stack?
* Q.15 What is p1\_ptr pointing to now? (Has it changed?)
* Q.16 Is the current value of p1\_ptr good or bad? Explain

**Section 5**

* Q.17 Uncomment the next code line - will it compile?
* Q.18 Does your IDE tell you of any issues? If so, how?
* Q.19 MAGIC NUMBER?! What is it? Is it bad? Explain!
* Q.20 Explain in your own words how the array size is calculated.
* Q.23 Change the size argument to 10 (or similar). What happens?
* Q23.1 You might see some values that we set earlier. Why would this happen?

**Section 6**

* Q24 Points to nothing - does it?
* Q.25 What is "hex" and what does it do? (url in your notes)
* Q.26 What is new and what did it do?
* Q.27 What is delete and what did it do?
* Q.28 What happens when we try this? Explain.
* Q.29 What is the difference between NULL and nullptr and 0?
* Q.30 What happens in this line? (A zero address now, so ...)

**Section 7**

* Q.31 Are default pointer values in an array safe? Explain.
* Q.32 We should always have "delete" to match each "new". What is the
  + problem if we don't delete, and what is the common name for this?
* Q.33 Your IDE may have tools to help you track memory. Does it?
* Q.34 Can you see what happens if you DON'T do this?
* Q.35 Should we set pointers to nullptr? Why?
* Q.36 How do you create an array with new and set the size?

**Out of the main**

Q.21-22 Go to the showParticleArray 2 implementation and see there:

* Q.21 What is the difference between this function signature and the function signature for showParticleArray?
* Q.22 Uncomment the following. It gives different values to those we saw before so it won't work as a way to determine array size - but why?