Task 5.P - 100595153

1. [line 55] What is the difference between a struct and a class?

The key difference between a struct and class ignoring the naming, is the default accessibility of each member variable and function. Classes are set to private, structs are set to public.

1. [line 63] What are function declarations?

Functions declarations are forward declarations of functions that are yet to be explicitly defined. These are generally used in header files, though as is clearly with the implementation here, can be done within cpp files too. The purpose of a forward declaration is to let the compiler know that the function exists so that declaration issues can be avoided.

1. [line 67] Why are variable names not needed here?

As these are forward declarations, the actual name of the parameters does not need to be included yet, nothing has referenced the parameters yet. This can be included during the function definition.

1. [line 75] Does your IDE know if this method is used?

Yes. Generally unused methods and variables are duller than those that are used. Additionally, the IDE sometimes shows a warning underline.

1. [line 86] un-initialised values ... what this show and why?

Currently, the variable has not been initialised. It has a type, but currently is null. There is no assigned value.

1. [line 95] Did this work as expected?

Without fixing the error from the previous question, particle is not output to console and instead, an error is thrown. After fixing the error, the particle is output and runs as expected.

1. [line 97] Initialisation list - do you know what are they?

I’m not the most familiar with them, but I believe I have used them at least once prior. They are essentially lists of values to be assigned to the class’s member variables on initialisation. As noted, having a list of variables is a tad easier than assigning one by one. Although, using the new keyword and initializing via constructor would probably be fine too, from what I understand.

1. [line 113] Should show age=1, x=1, y=2. Does it?

No… It should not… I assume this question contains typos.

1. Q.9 [line 117] Something odd here. What and why?
2. Q.10 [line 128] showParticle(p1) doesn't show 5,6,7 ... Why?
3. Q.11 [line 153] So what does -> mean (in words)?
4. Q.12 [line 154] Do we need to put ( ) around \*p1\_ptr?
5. Q.13 [line 160] What is the dereferenced pointer (from the example above)?
6. Q.14 [line 165] Is p1 stored on the heap or stack?
7. Q.15 [line 166] What is p1\_ptr pointing to now? (Has it changed?)
8. Q.16 [line 172] Is the current value of p1\_ptr good or bad? Explain
9. Q.17 [line 175] Is p1 still available? Explain.
10. Q.18 [line 180] <deleted - ignore> :)
11. Q.19 [line 189] Uncomment the next code line - will it compile?
12. Q.20 [line 192] Does your IDE tell you of any issues? If so, how?
13. Q.21 [line 200] MAGIC NUMBER?! What is it? Is it bad? Explain!
14. Q.22 [line 207] Explain in your own words how the array size is calculated.
15. Q.23 [line 375] What is the difference between this function signature and
16. Q.24 [line 380] Uncomment the following. It gives different values to those we saw before
17. Q.25 [line 219] Change the size argument to 10 (or similar). What happens?
18. Q.26 [line 237] What is "hex" and what does it do? (url in your notes)
19. Q.27 [line 242] What is new and what did it do?
20. Q.28 [line 252] What is delete and what did it do?
21. Q.29 [line 256] What happens when we try this? Explain.
22. Q.30 [line 265] So, what is the difference between NULL and nullptr and 0?
23. Q.31 [line 267] What happens if you try this? (A zero address now, so ...)
24. Q.32 [line 302] Are default pointer values in an array safe? Explain.
25. Q.33 [line 317] We should always have "delete" to match each "new".
26. Q.34 [line 325] Should we set pointers to nullptr? Why?
27. Q.35 [line 330] How do you create an array with new and set the size?