

ex00

There must be a `ponyOnTheHeap` function that allocates a new Pony using "new", then deletes it (using "delete", obviously).

There must be a `ponyOnTheStack` function that allocates a new Pony on the stack (WITHOUT using "new" or "malloc").

✓ Yes

✗ No

ex01

ex01

Theoretically, two choices here are right: Either changing the allocation to be on the stack and not on the heap (So, don't use "new" anymore, and handle a `std::string` without using the pointer), or adding a "delete panthere;" after the `std::cout`.

Ask the student to explain WHY he did what he did before marking this as done. He has to answer something other than "Meh, it just works".

✓ Yes

✗ No

ex02

As the subject says, this exercise requires the student to turn in a "main" function, and it must, when run, demonstrate that the exercise works as intended (for example, using console output, etc...). The student has to be able to explain how it proves anything. If it does not, you MUST NOT grade this part.

ex02

All the classes and functions required by the subject must exist and work as specified, otherwise, no points for this exercise.

The Zombies must be destroyed when appropriate.

In `newZombie`, it should be allocated on the heap, returned, and then deleted in the `main()`. The student must explain why.

The Zombies created by `randomChump` must either be allocated on the stack (so implicitly deleted at the end of the function), or allocated on the heap then explicitly deleted.

The student must justify his choice.

ex03

As the subject says, this exercise requires the student to turn in a "main" function, and it must, when run, demonstrate that the exercise works as intended (for example, using console output, etc...). The student has to be able to explain how it proves anything. If it does not, you MUST NOT grade this part.

ex03

All the classes and functions required by the subject must exist and work as specified, otherwise, no points for this exercise.

The Zombies must be allocated in the constructor of the ZombieHorde, and should be allocated as an array, either on the stack, either explicitly using `new[]`, in which case they should be deleted in the destructor. The student must explain his choice.

 Yes

 No

ex04

ex04

There is a string containing "HI THIS IS BRAIN", then a pointer to it, then a reference to it, and it is displayed through the pointer then through the reference. As the subject says, really, that's it, no tricks or anything.

 Yes

 No

ex05

As the subject says, this exercise requires the student to turn in a "main" function, and it must, when run, demonstrate that the exercise works as intended (for example, using console output, etc...). The student has to be able to explain how it proves anything. If it does not, you MUST NOT grade this part.

ex05

All the classes and functions required by the subject must exist and work as specified, otherwise, no points for this exercise.

The "identify" function

in the "Brain" must return the representation of "this", or any other trick that equates to "the adress of the current instance".

The "getBrain"

function should return a REFERENCE to the Brain of the current Human. With the main() that the subject provides, it must, as the subject says, display two identical addresses.

The student should be able to explain why he did this.



ex06

As the subject says, this exercise requires the student to turn in a "main" function, and it must, when run, demonstrate that the exercise works as intended (for example, using console output, etc ...). The student has to be able to explain how it proves anything. If it does not, you MUST NOT grade this part.

ex06

All the classes and functions required by the subject must exist and work as specified, otherwise, no points for this exercise.

The student must choose to store the Weapon either as pointer or as a reference in HumanA and HumanB.

In HumanA, BOTH are acceptable if justified, even if theoretically the reference is better, since the Weapon exists from creation until destruction and never changes.

In HumanB, only the pointer is acceptable, since the field is not set at creation time, so it can not be a reference.

The student must justify his choices correctly.



ex07

ex07

The program must work as the subject specifies. A reasonable amount of errors must be handled. If you can find an error that isn't handled, and isn't completely esoteric, no points for this exercise.

The program must read from the file using an ifstream or equivalent, and write using an ofstream or equivalent."

✓ Yes

✗ No

ex08

As the subject says, this exercise requires the student to turn in a "main" function, and it must, when run, demonstrate that the exercise works as intended (for example, using console output, etc...). The student has to be able to explain how it proves anything. If it does not, you MUST NOT grade this part.

ex08

All the classes and functions required by the subject must exist and work as specified, otherwise, no points for this exercise.

The "action" function must use an array of pointer to member functions to choose which action should be called. Any if/elseif/elseif/else or other crap like this counts as wrong.

✓ Yes

✗ No

ex09

As the subject says, this exercise requires the student to turn in a "main" function, and it must, when run, demonstrate that the exercise works as intended (for example, using console output, etc...). The student has to be able to explain how it proves anything. If it does not, you **MUST NOT** grade this part.

ex09

All the classes and functions required by the subject must exist and work as specified, otherwise, no points for this exercise.

Must work exactly as the subject requires.

As with the previous exercise, the action to take when using "log" must be determined using an array of pointers to member functions.

✓ Yes

✗ No

ex10

ex10

The program must work as the subject specifies. Any error that isn't handled = no points for this exercise."

✓ Yes

✗ No

Ratings

Don't forget to check the flag corresponding to the defense