Workshop 7 reflections

1. Why did we choose m_health to be a signed integer? (Hint: what happens if the Hero has a health of 1, and someone does 2 points of damage to them.)

The reason for negative numbers is that if health is unsigned, then health is always > 0. The Hero will never die. Since the health can never be -1, the number will overflow and become very large. On a modern Intel PC, it will become (1 – 2) == 4294967295.

https://msdn.microsoft.com/en-us/library/7fh3a000.aspx

INT_MIN	Minimum value for a variable of type `int`.	–2147483647 – 1
INT_MAX	Maximum value for a variable of type `int`.	2147483647
UINT_MIN	Minimum value for unsigned int	0
UINT_MAX	Maximum value for a variable of type `unsigned int`.	4294967295 (0xffffffff)

Note the original Pacman game had an integer overflow bug. The integer they used was 8-bits, so
It overflowed after you played level 256.
http://errors.wikia.com/wiki/Pac_Man_-_Infamous_Kill_Screen_Bug

2. Does the Hero class need to know about the existence of the class SuperHero? (Hint: do a search in Hero.cpp, does the word "SuperHero" appear anywhere in it?) How about the reverse, does SuperHero know about the Hero class?

Parent (Base) classes are created before Child (Derived) classes, so they do not need to know about their existence. This is a feature.

Note: Hero is not a "friend class" of SuperHero, it is a Base class for SuperHero. Using friend class is frowned upon, since it gives another class full access to your members.

3. The program prints out "AncientBattle!" when 2 Heros fight. It prints out "SuperFight!" when 2 SuperHeros fight. When you tried to make a Hero fight a SuperHero, what did it print out?

You can, C++ will convert a SuperHero to a Hero and make it into a Hero vs Hero fight. So it prints out "AncientBattle."

Another way to look at it is this: our C++ program has the following functions

```
Hero&         fight (Hero&,         Hero&)
SuperHero&    fight (SuperHero &,   SuperHero&)
```

but not these:

```
Hero&         fight (Hero &,        SuperHero&)
Hero &        fight (SuperHero&,    Hero&)
```

Since it cannot convert a Hero into a SuperHero (remember question 2!), instead it will convert a SuperHero to a Hero and pass the values to Hero& fight (Hero&, Hero&).

It’s not the proper way to make a real videogame, but it shows the behaviour of classes well.

4. ***True or False: are the following a valid program definition for main?***

Yes, it is. It uses temporary objects (so avoids cluttering up the code with throwaway variable names). And it has a nice structure

Note: main() is the only C++ function which is allow to exit without a return statement. If control reaches the end of main, "return 0" is executed.