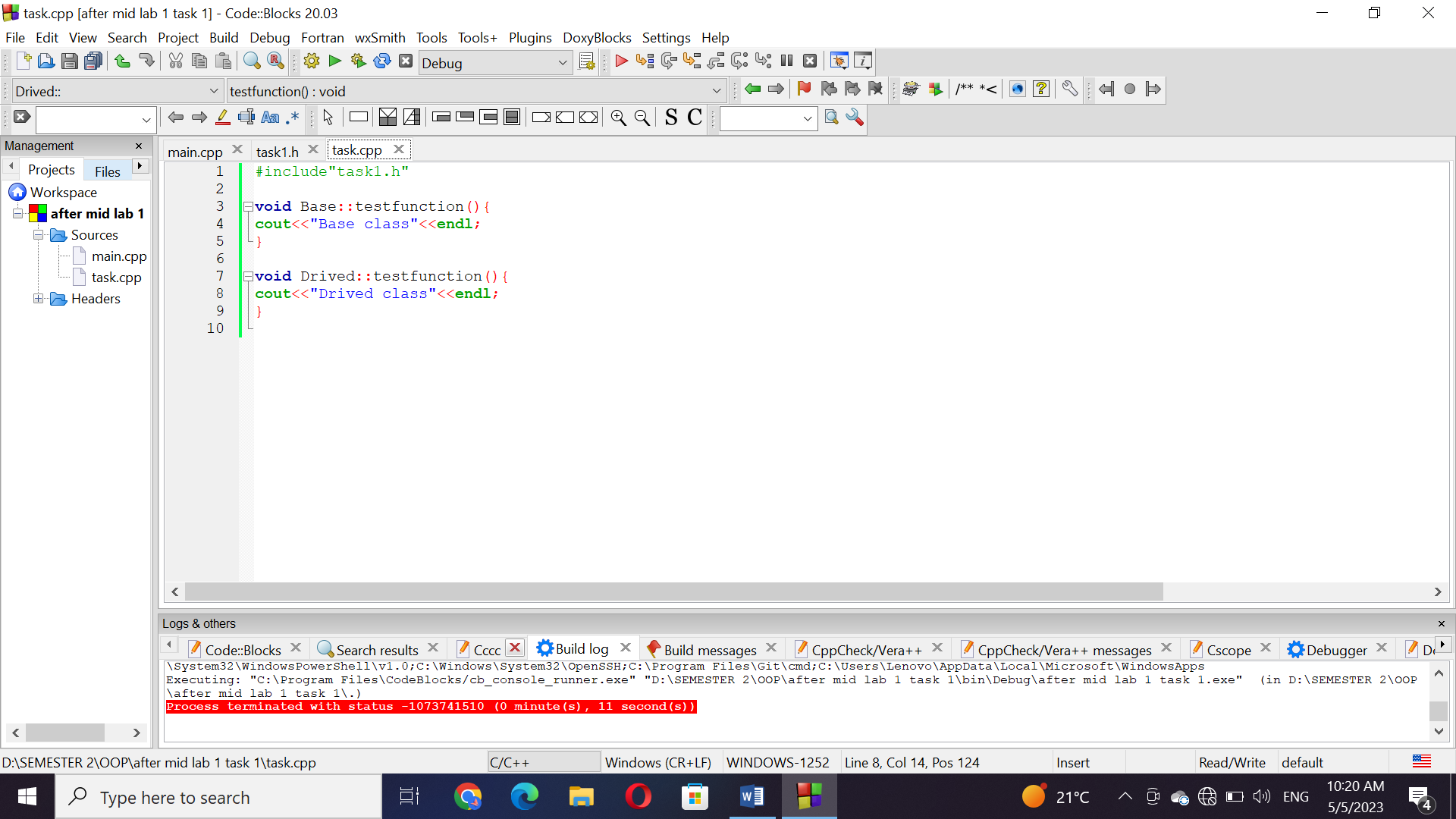
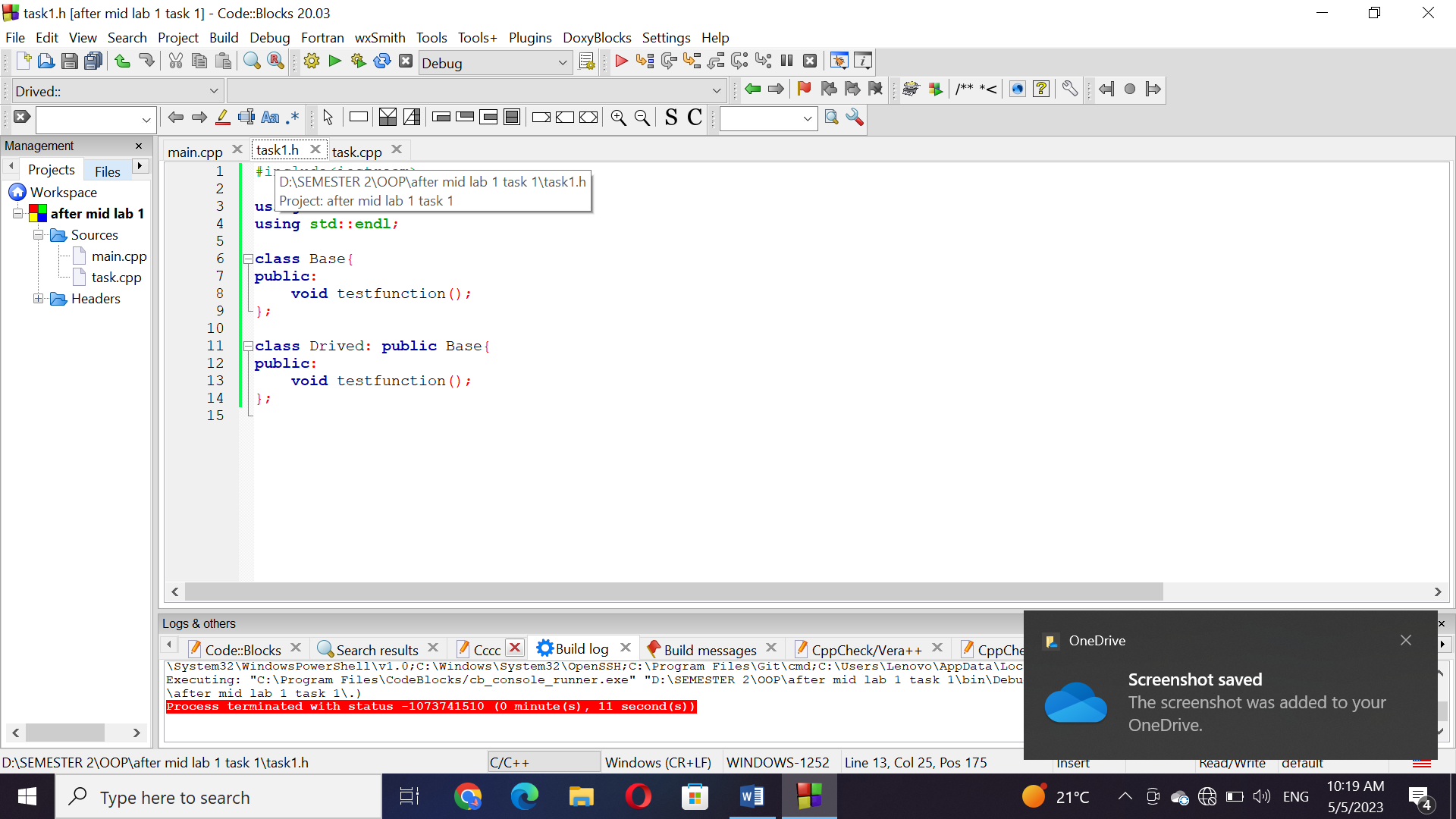
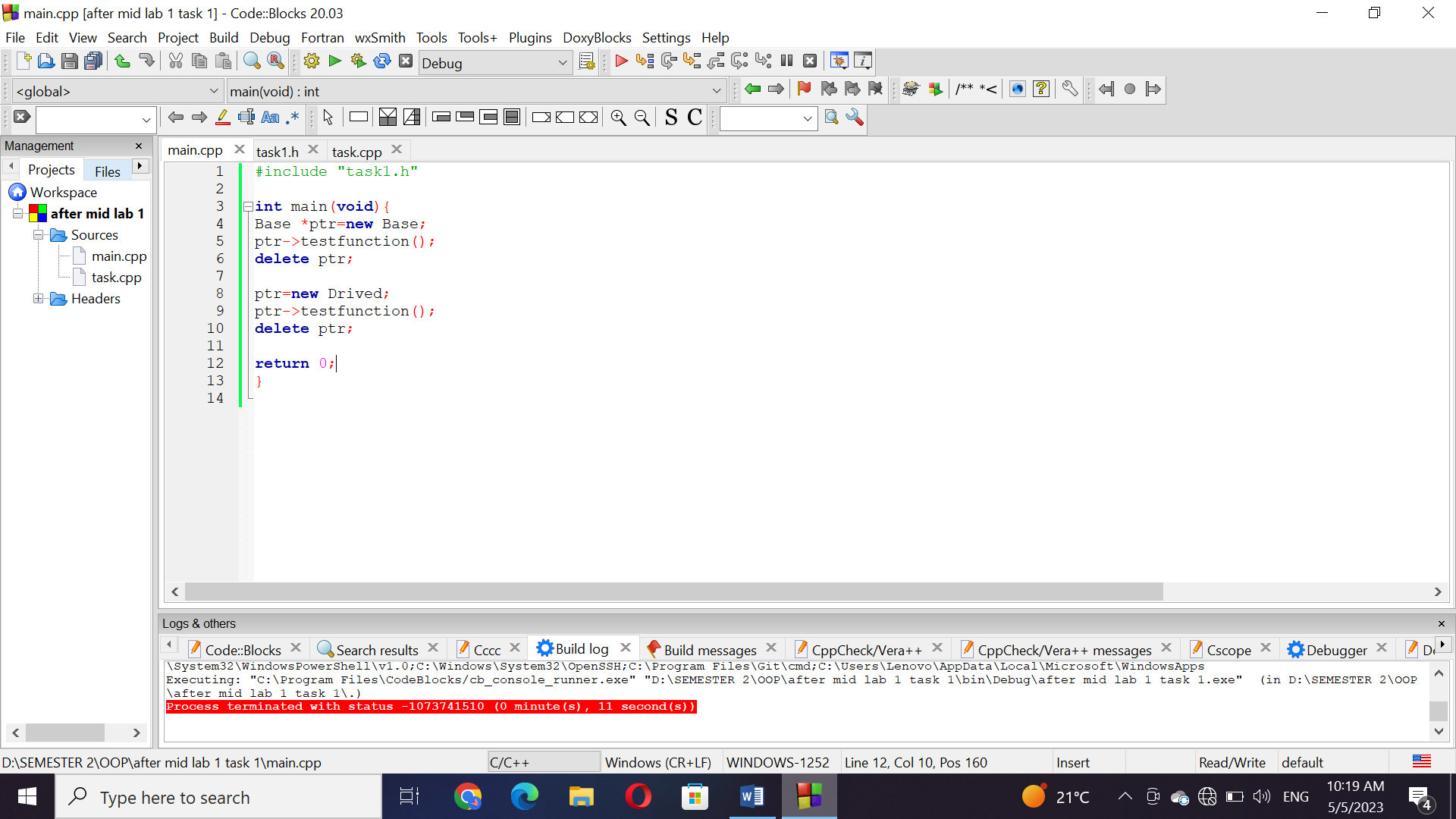
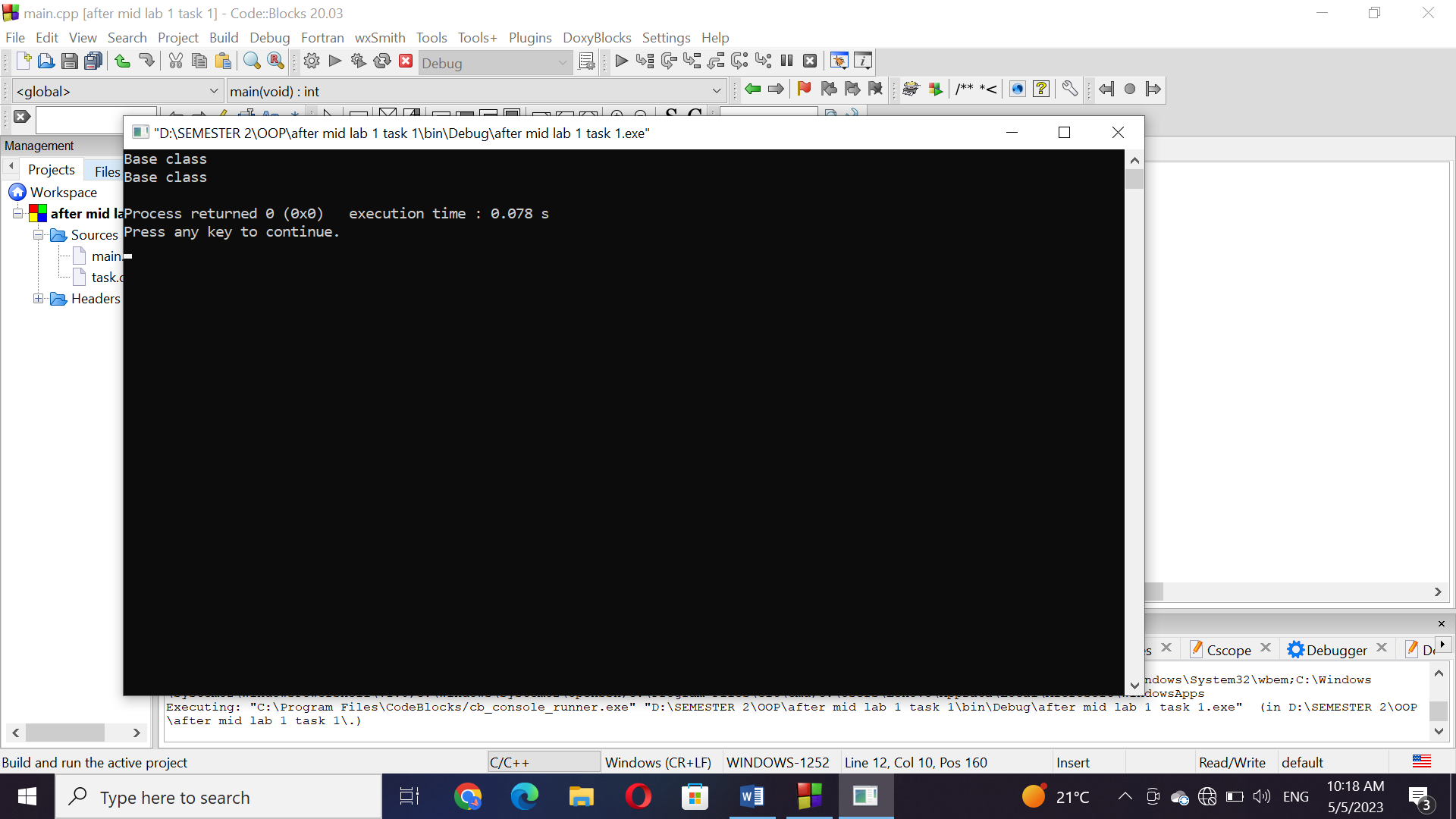
Task#1:

When we compile the code without using virtual keyword. The output was:

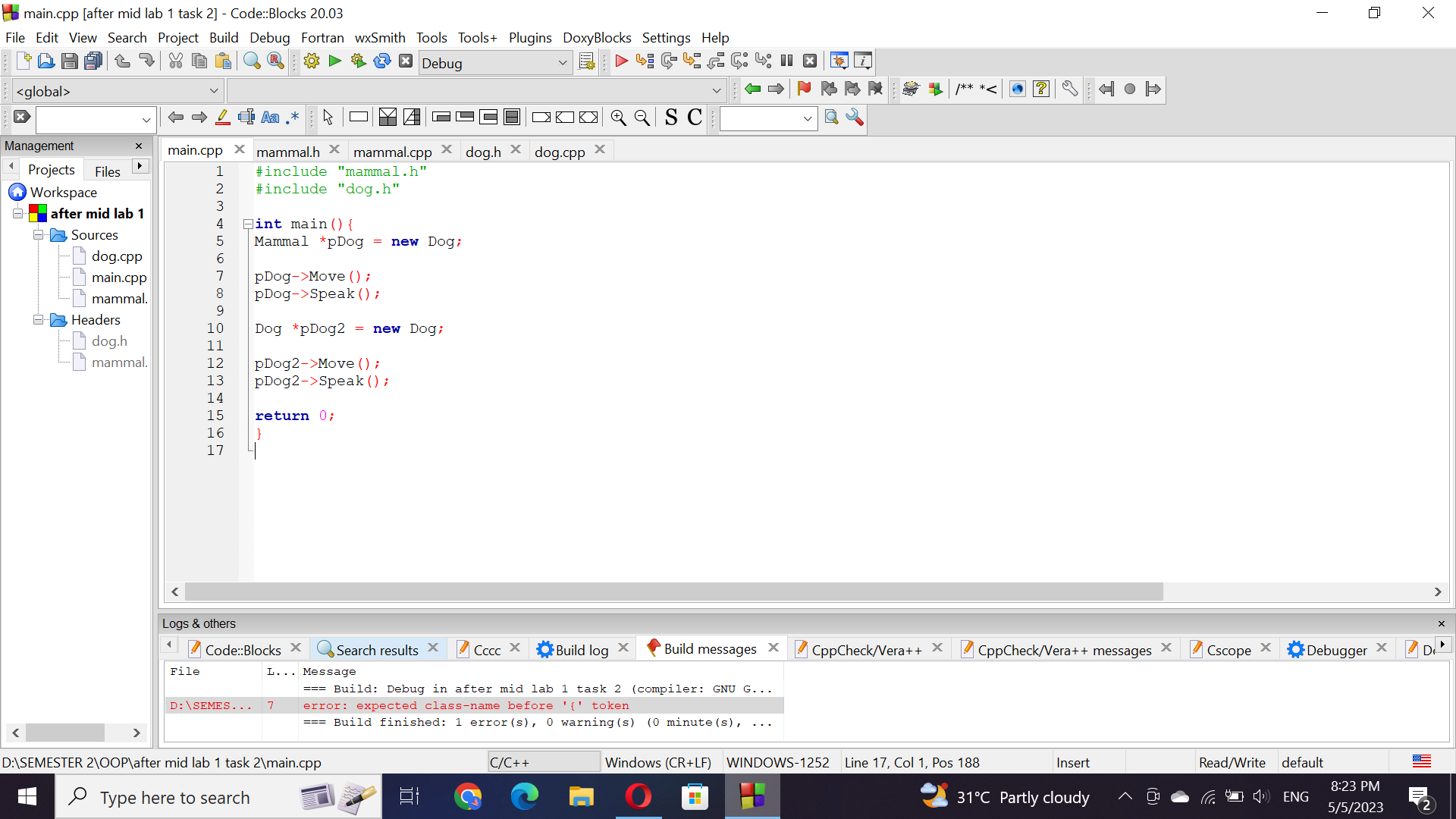
Base class

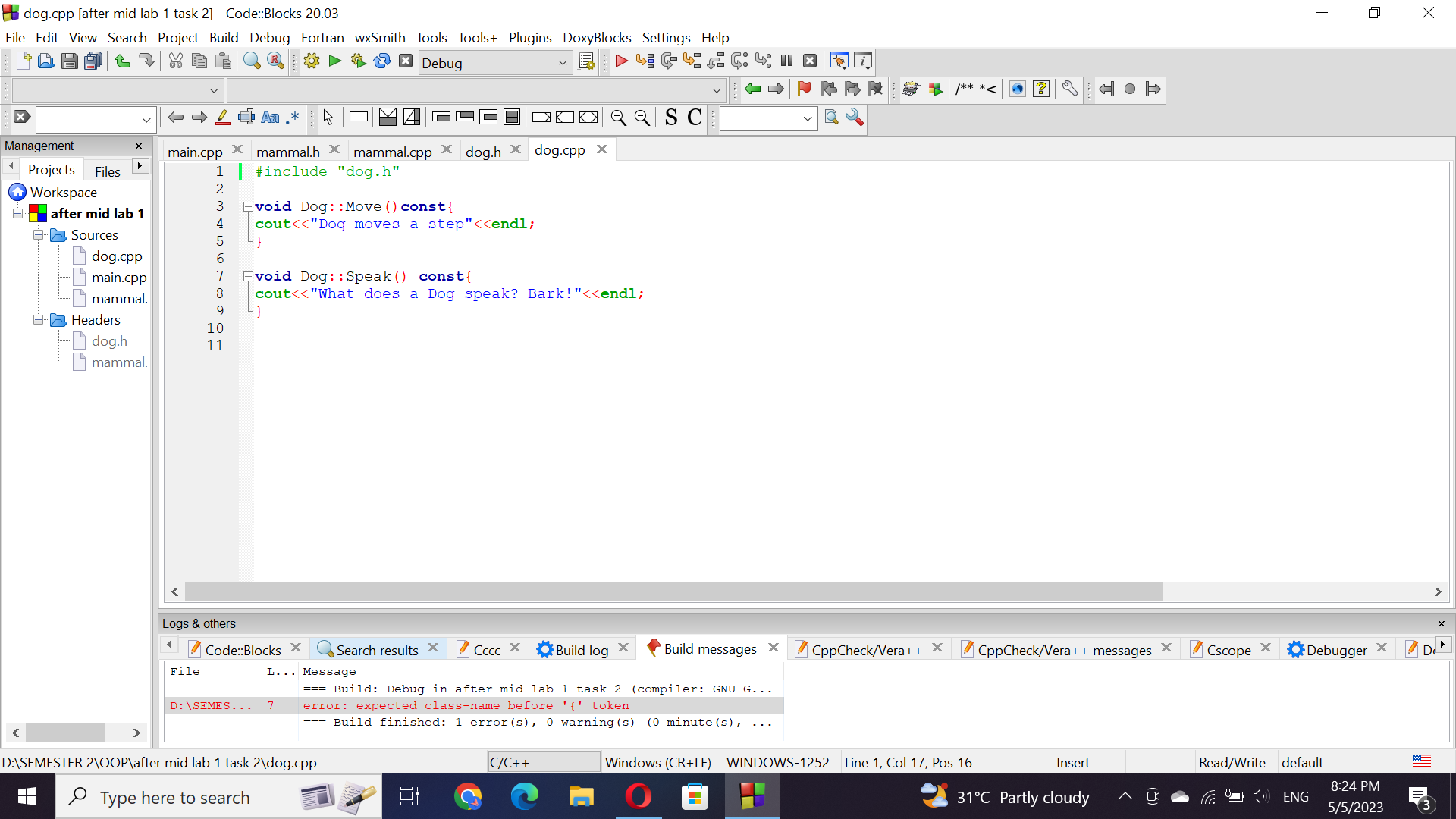
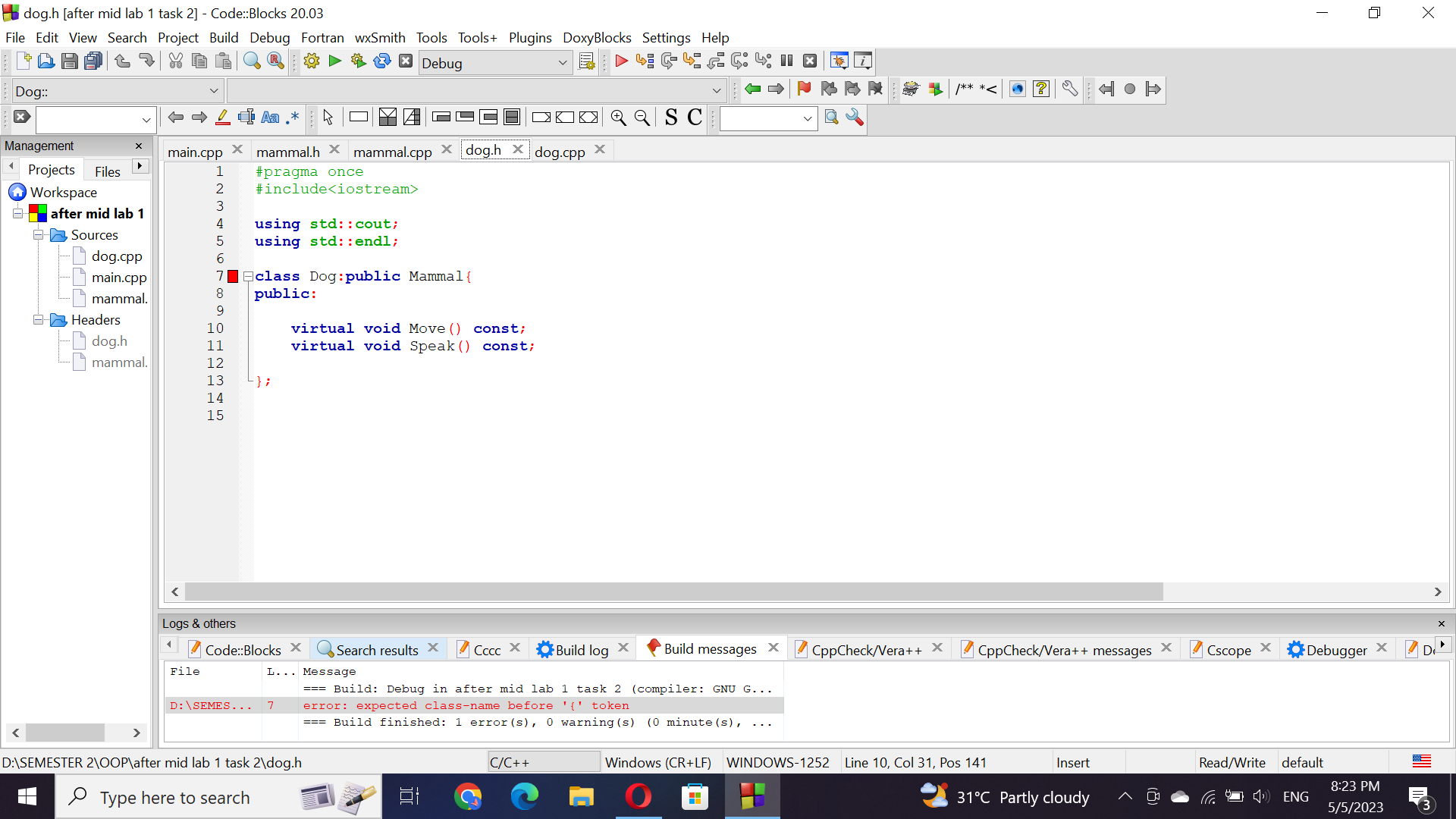
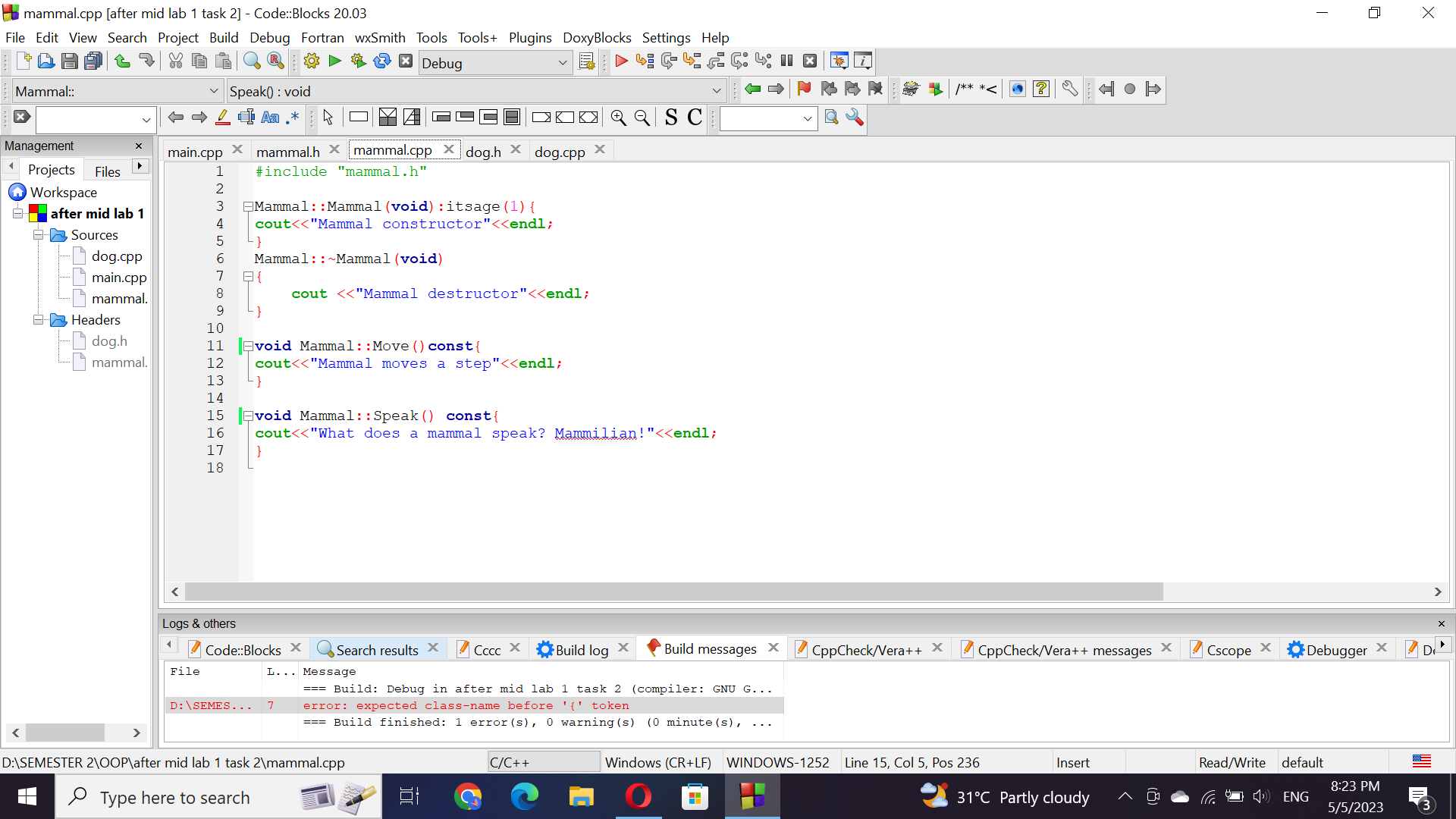
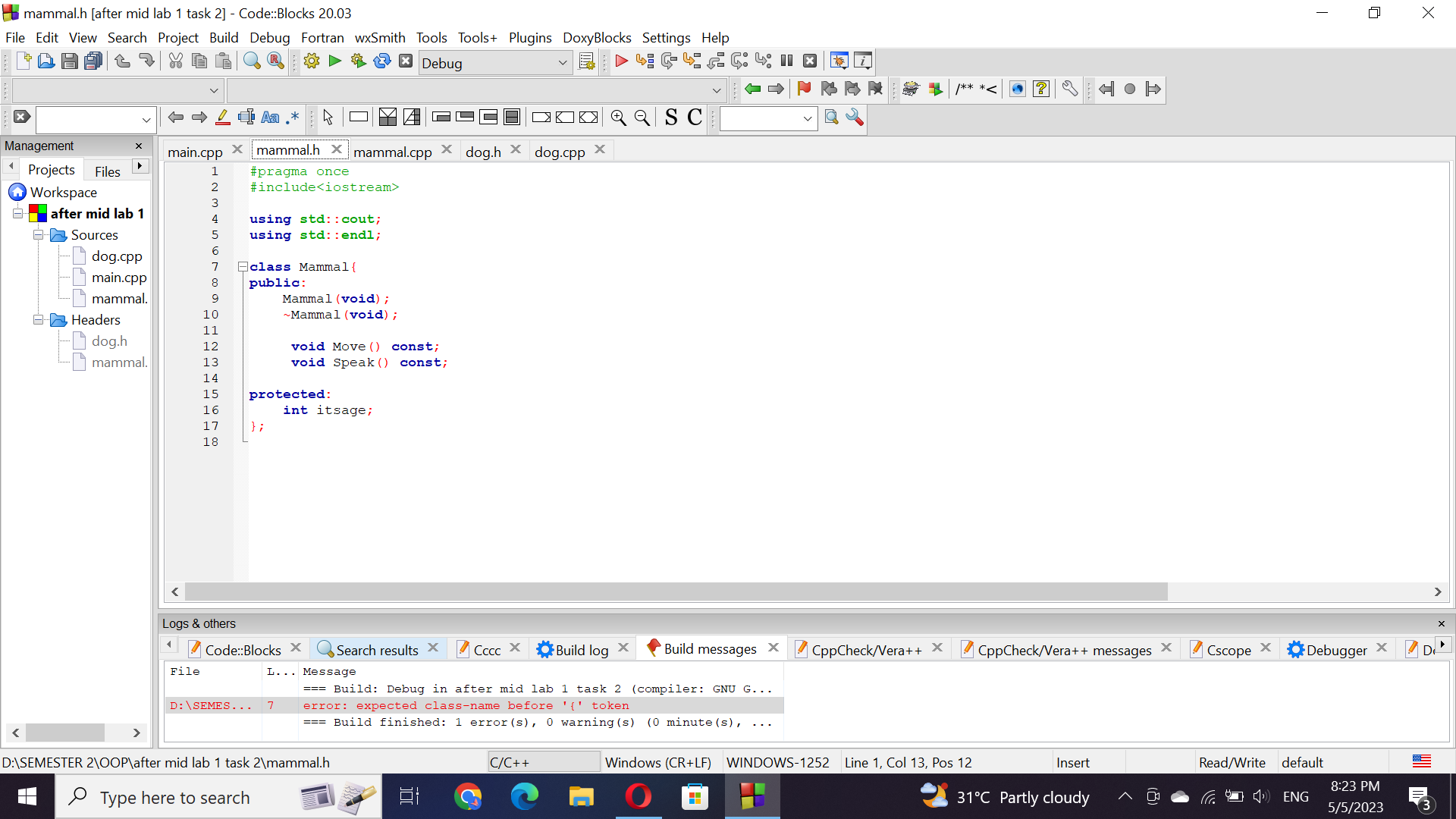
Base class

And after compiling the code with virtual keyword there was an error that “the base class was declared again”.

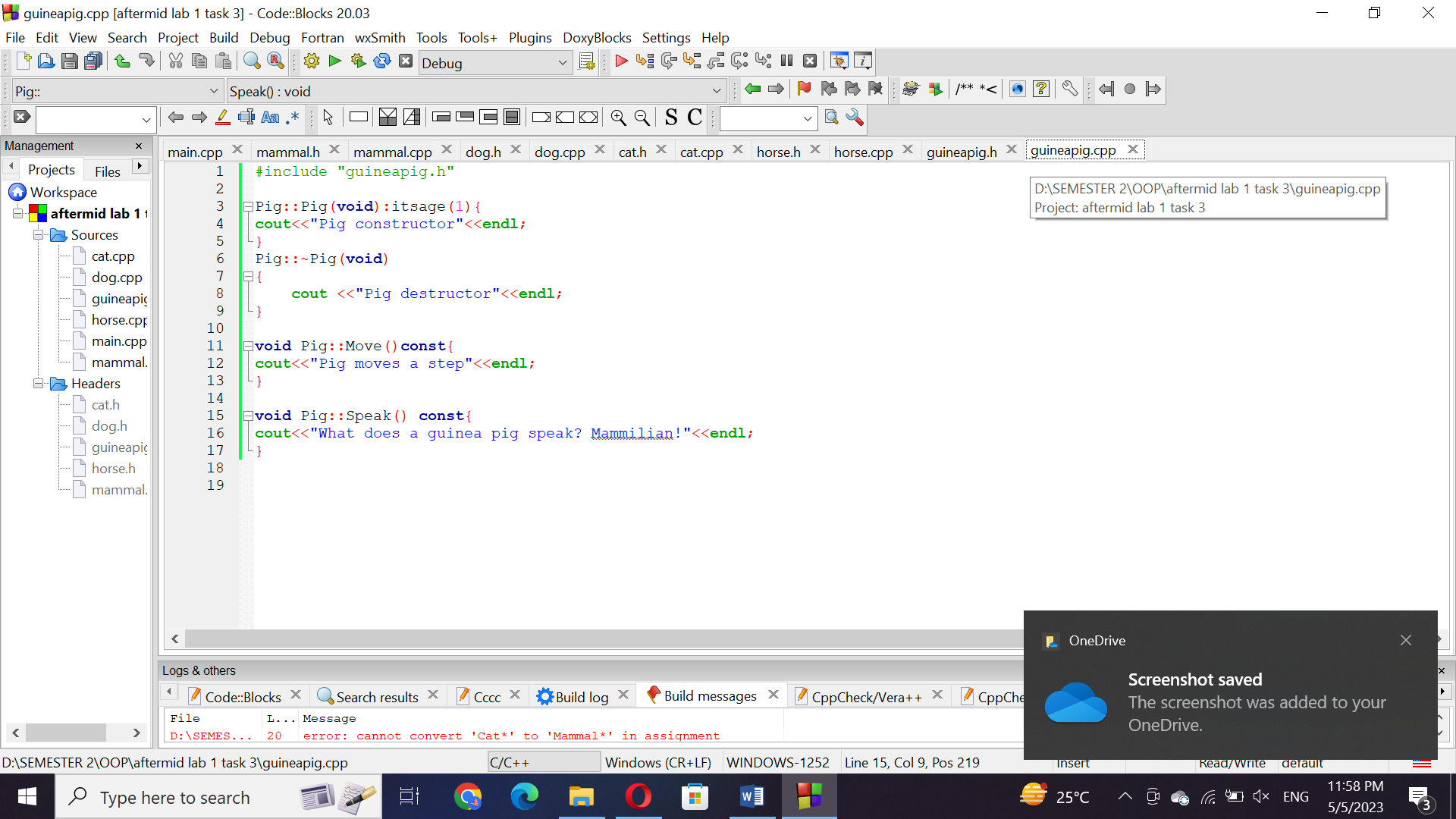
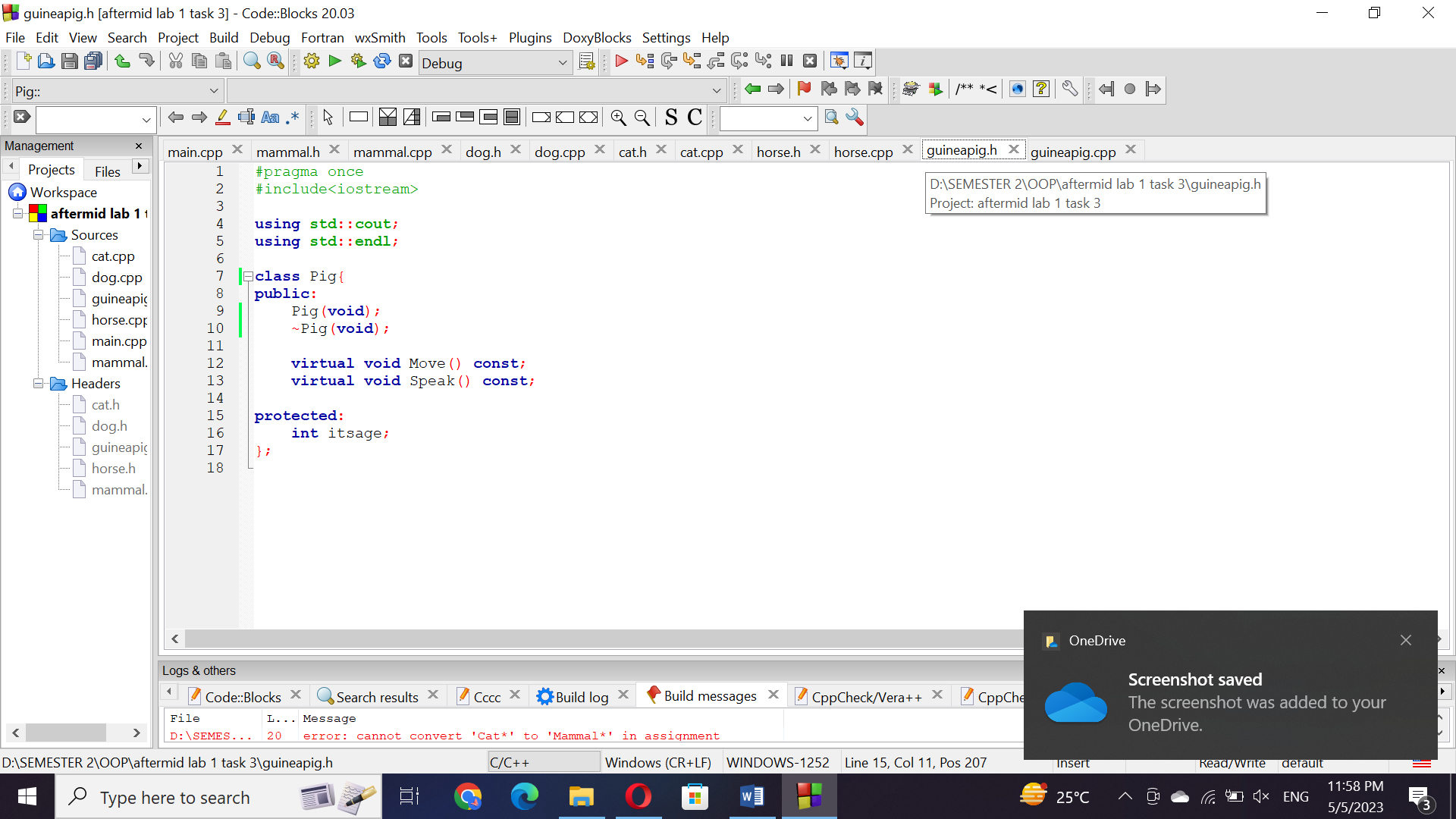
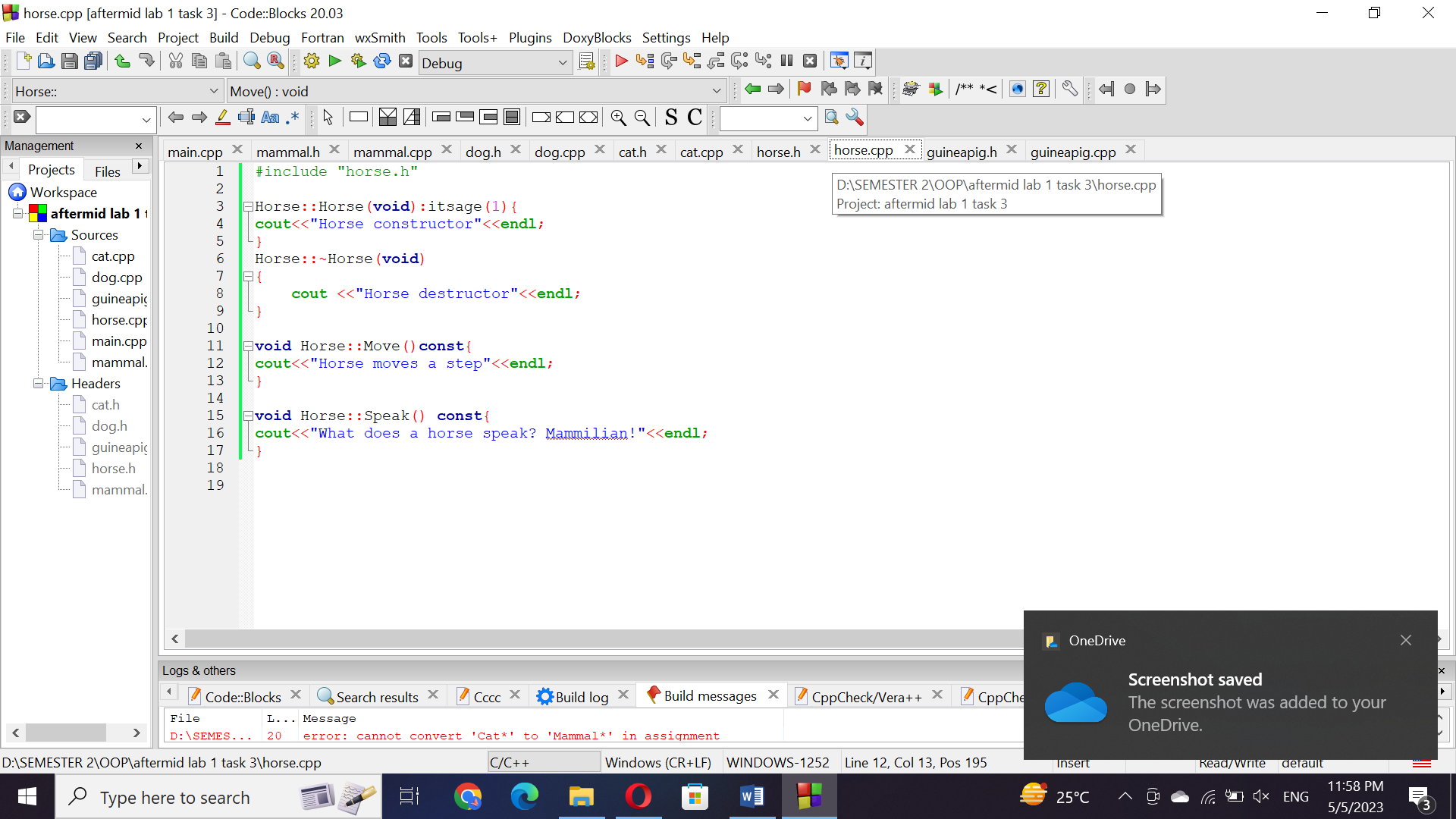
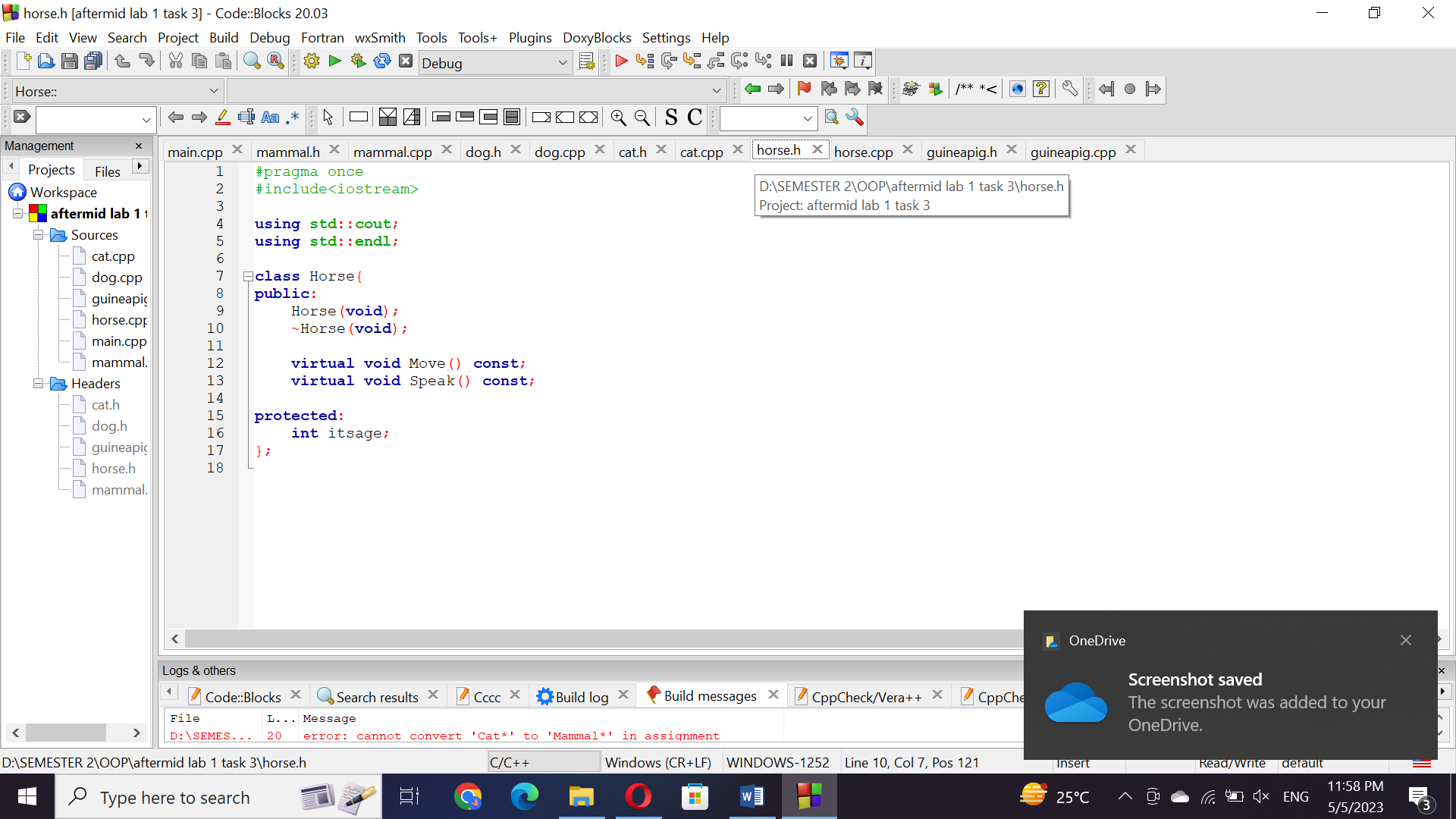
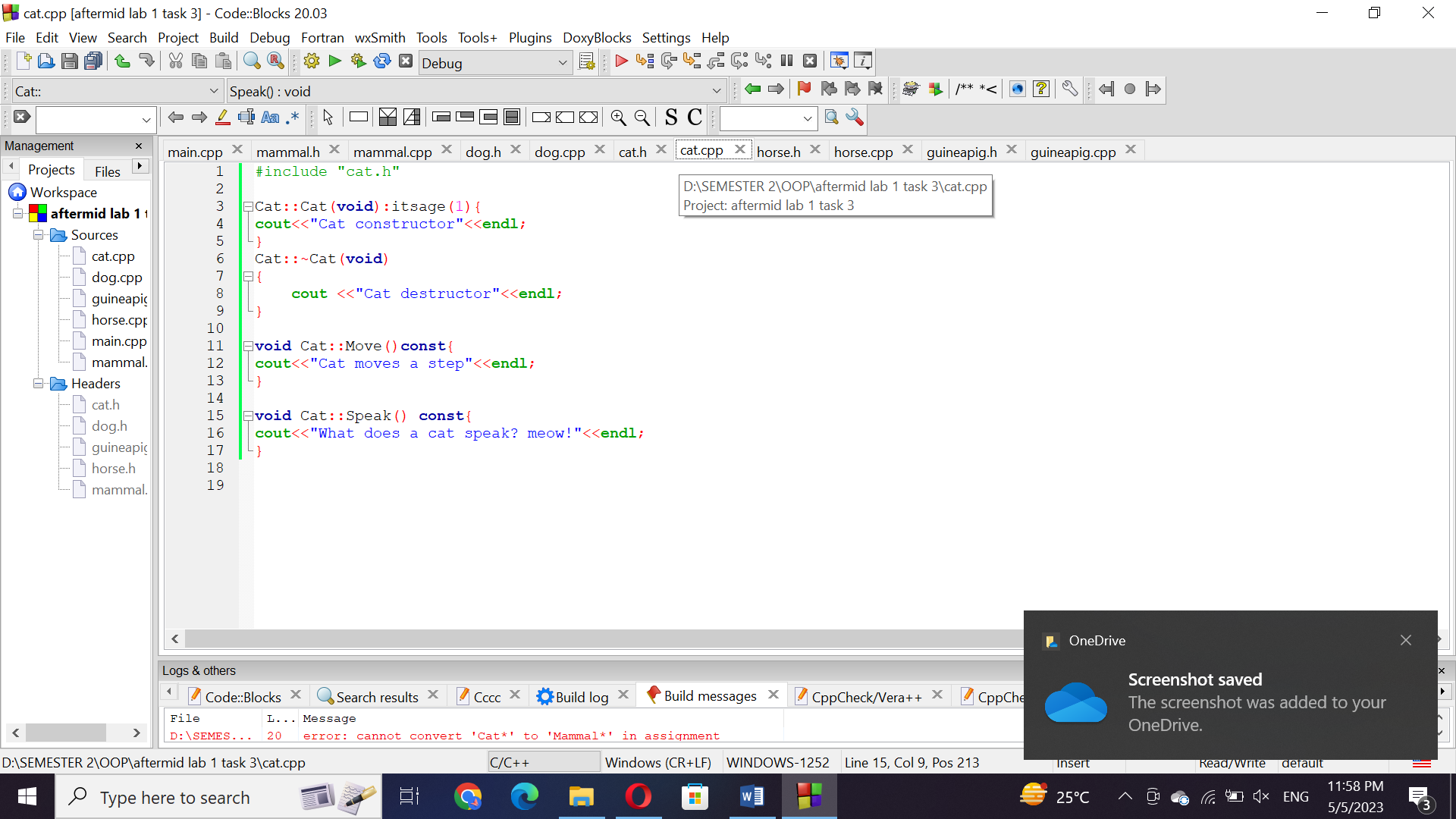
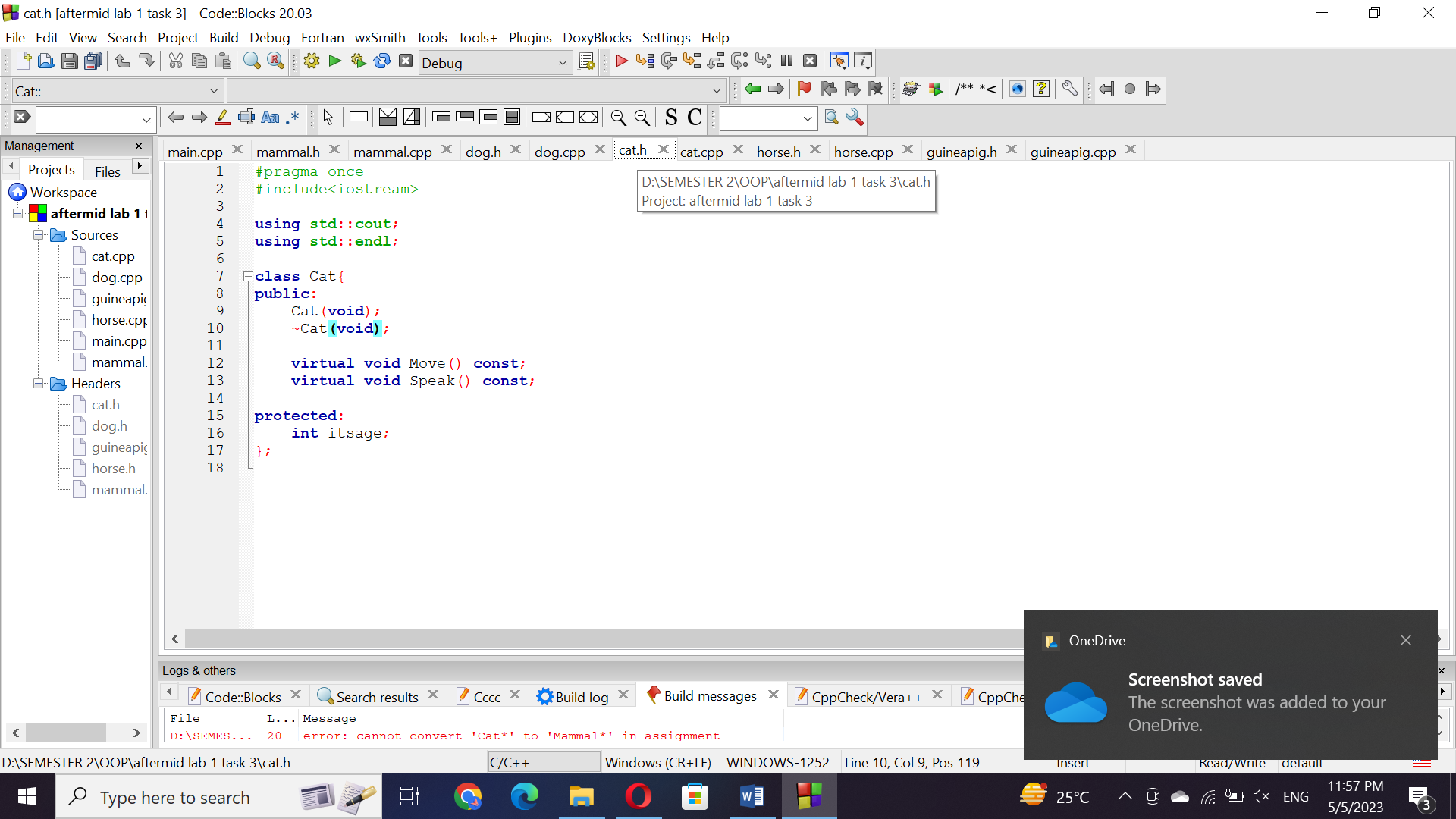
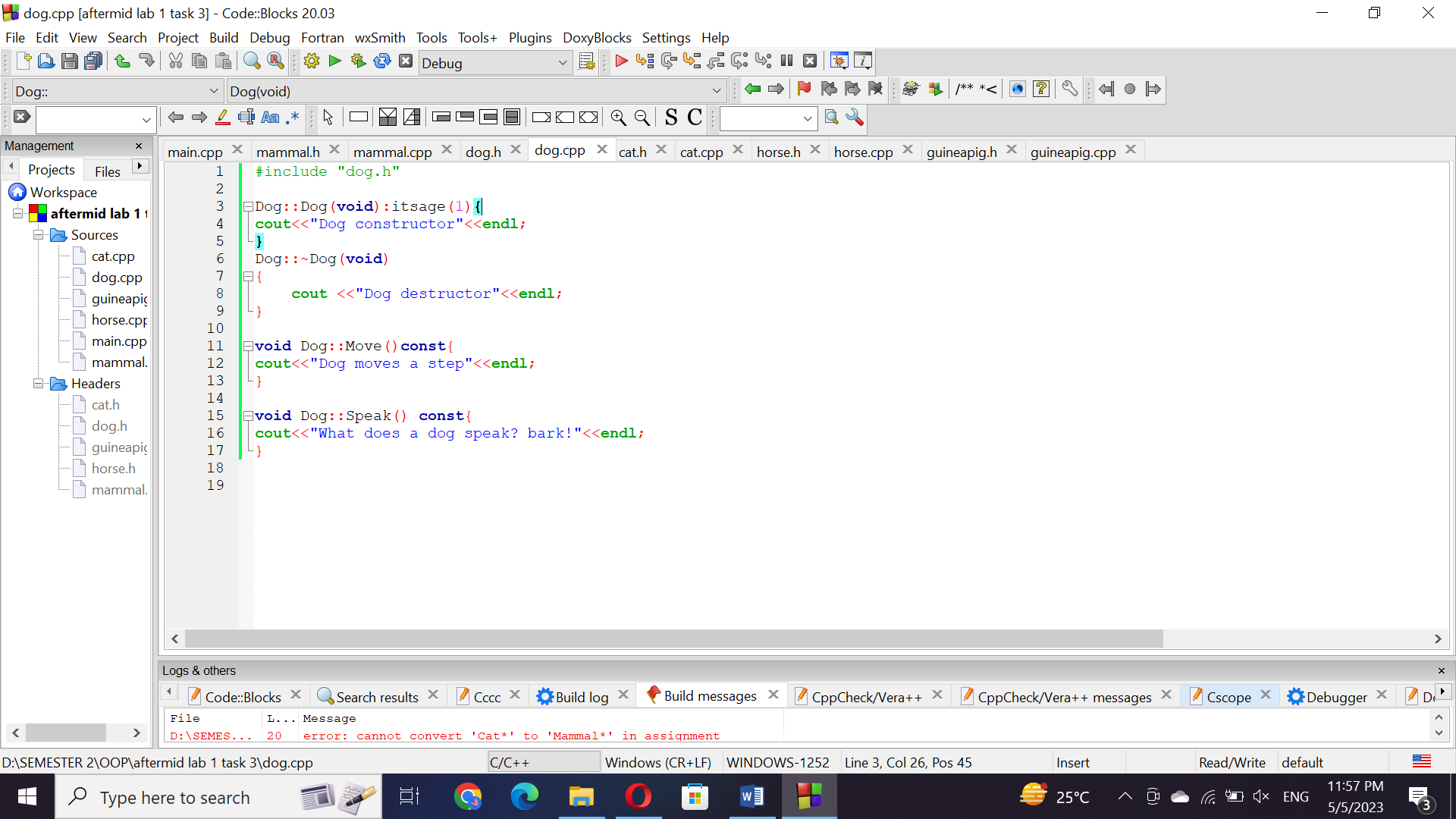
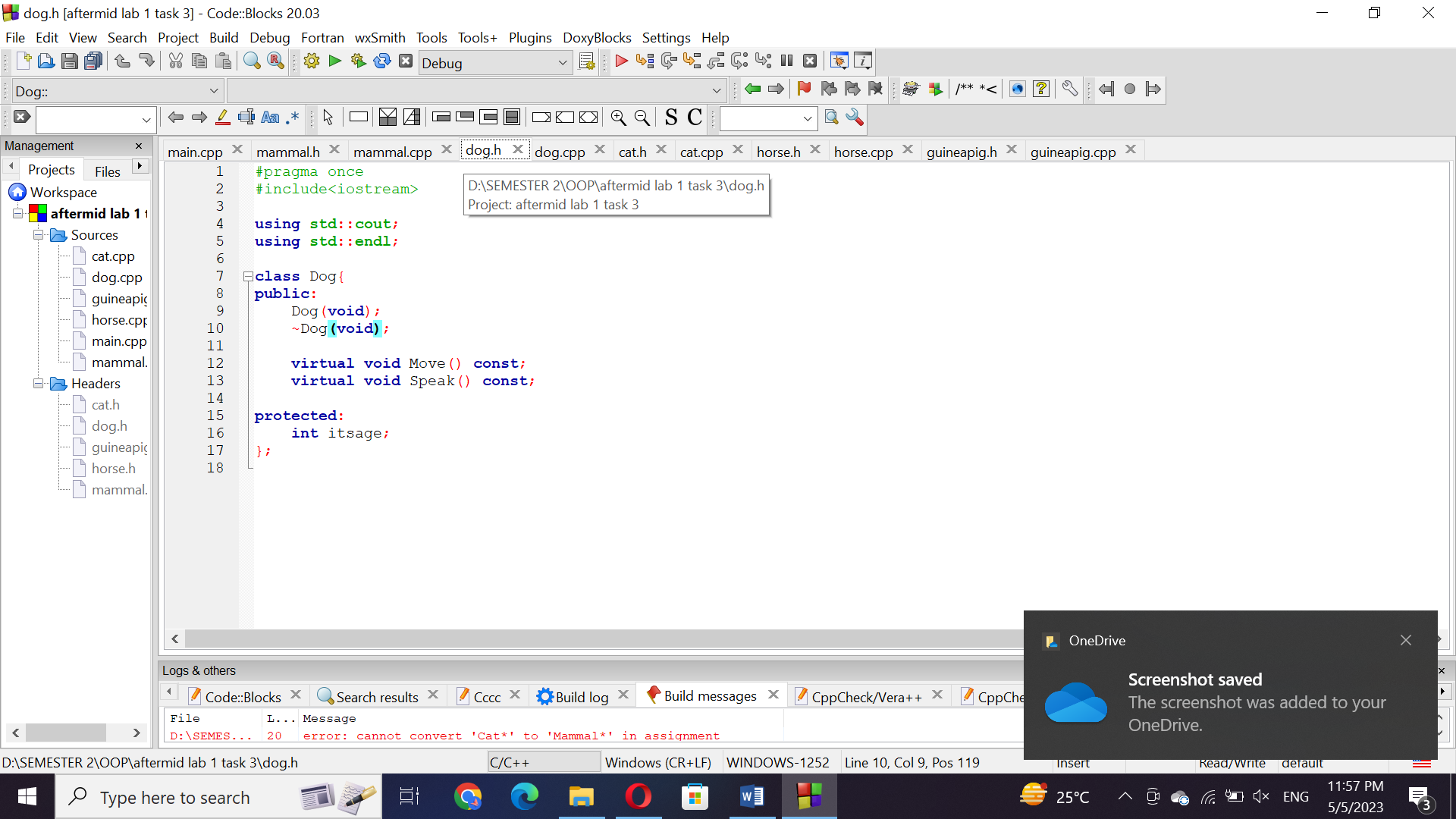
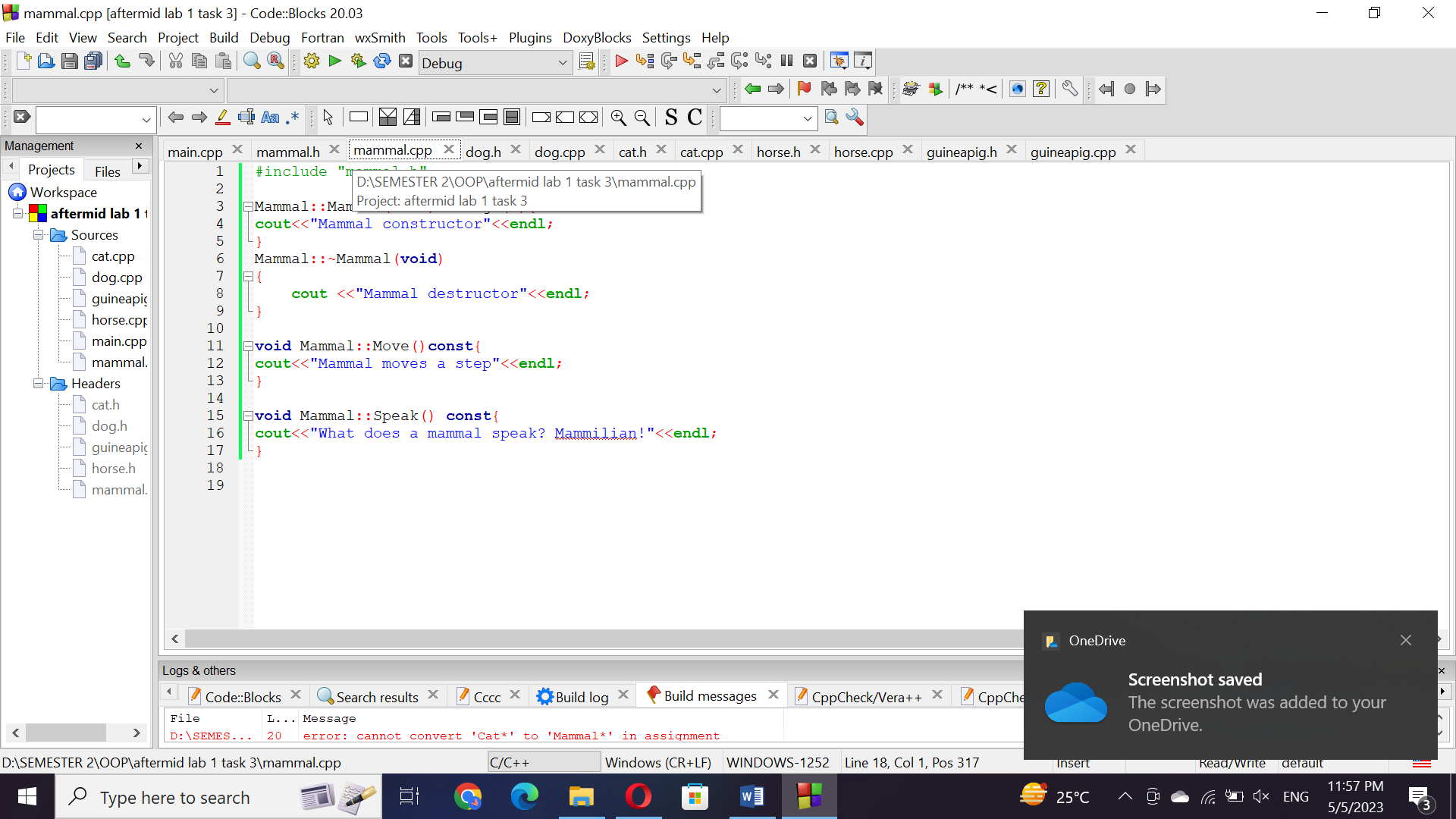
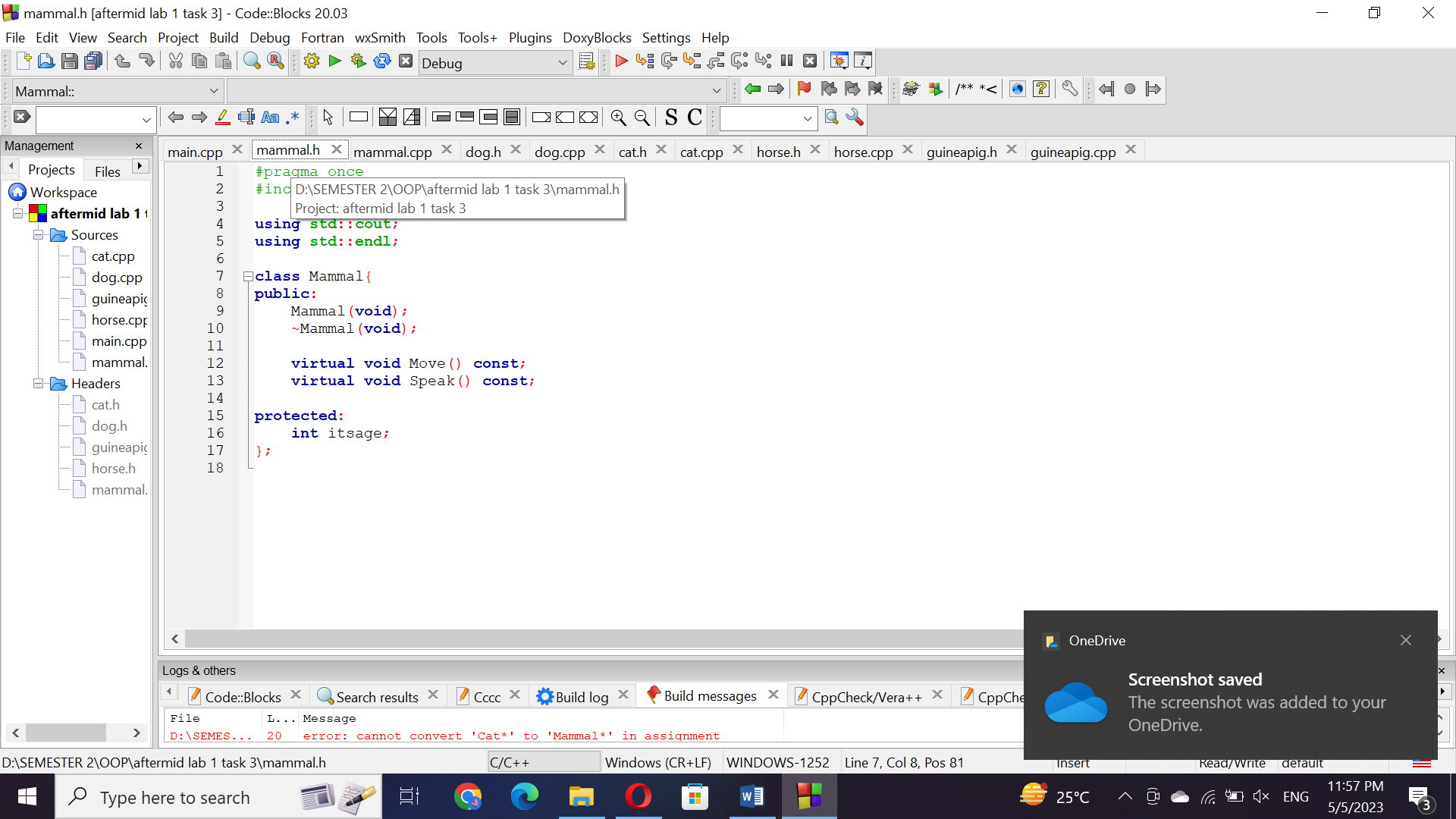
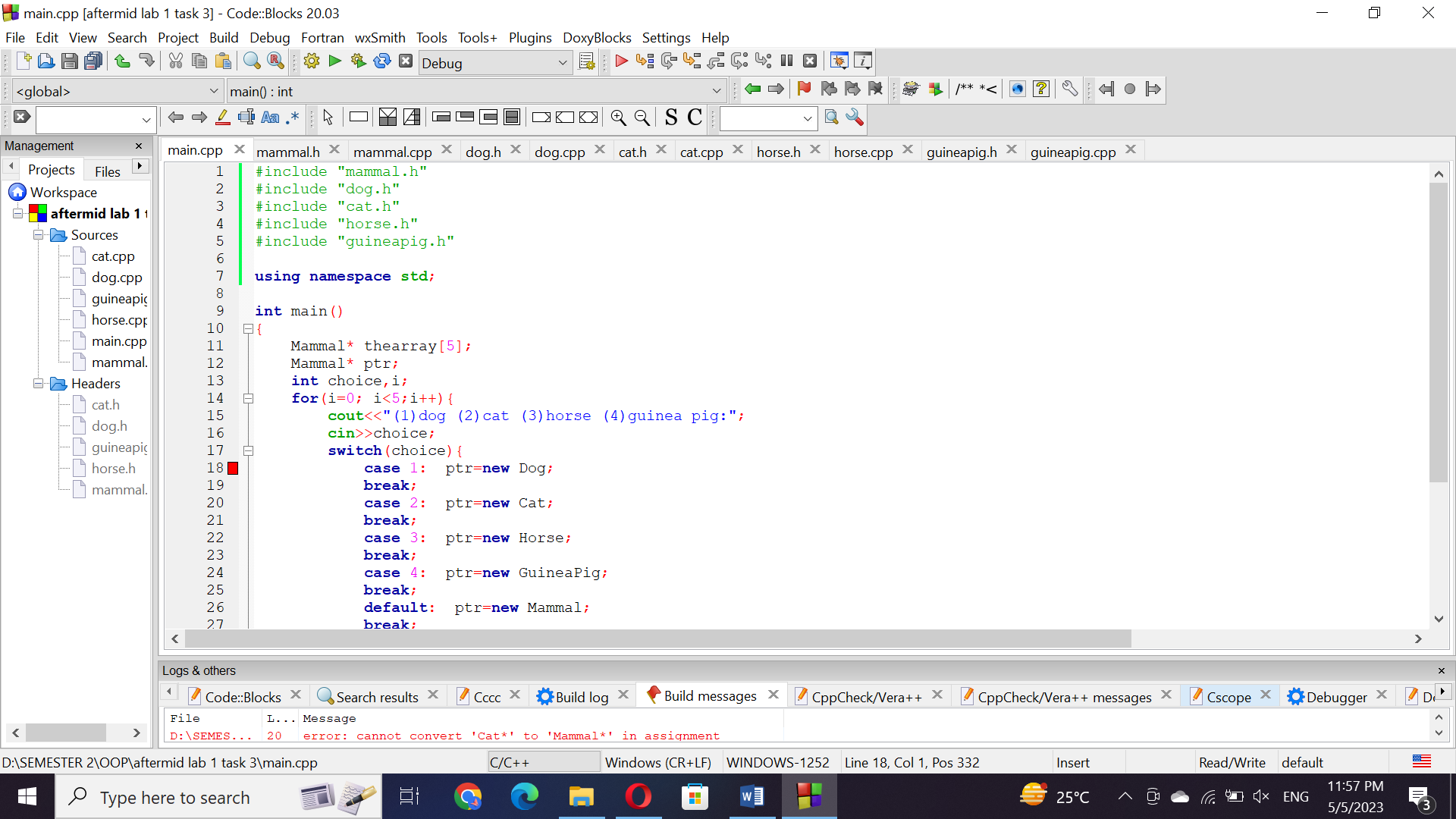


Task#2:





Task#3:



Questions:

Are inherited members and functions passed along to subsequent generations? If Dog derives from Mammal, and Mammal derives from Animal, does Dog inherit Animal's functions and data?

Yes, dog will inherit anumals functions and data.

Q. If, in the example above, Mammal overrides a function in Animal, which does Dog get, the original or the overridden function?

Q. Can a derived class make a public base function private?

No, a derived class cannot make a public base function private.

Q. Why not make all class functions virtual?

We can make all classes virtual but it is not efficient because it will take more memory. It can make the code harder to understand and maintain.

Q. If a function (SomeFunc()) is virtual in a base class and is also overloaded, so as to take either an integer or two integers, and the derived class overrides the form taking one integer, what is called when a pointer to a derived object calls the two-integer form?

If a function in a base class is virtual and is overloaded to take either an integer or two integers, and the derived class overrides the form taking one integer, then when a pointer to a derived object calls the two-integer form, the base class's two-integer form will be called.