OOD, BRIEFLY

A class should be self-contained. When you create a class you define its 'purpose' in your larger program. You identify the data required for that purpose. You identify the actions that need to be performed on that data. You look at those actions. Is it something that external classes must be able to do? Is it something that is just needed inside the class? After that review, you check if you need any additional auxiliary actions that would be restricted to use inside the class.

Now you can draft or outline your class. Data, with very, very few exceptions should be private. You need to decide which functions (defined from the required actions) are public or private. Then you have your base class.

COMMENT: What I've said applies to creating a class. For example, C++ allows you to use classes but the entire program does not need to be object oriented. To do full OOD we move on.

You do this for the classes you've identified. If you are considering inheritance the next step is to review the classes. Ignore the "semantics", i.e. what you know about the real situation modeled by the code. Look at each class and if there's overlap between 2 or more then you can create a parent class with the overlapped materials. Then repeat. Likewise, if you thought you needed to inherit and you realize the child class is using nothing from the parent, then there is no relation and there should be no inheritance.

This is the 64 cent tour of OOD. :-)