How does your design implement the four pillars of OOP (abstraction, encapsulation, inheritance and composition, and polymorphism)?

My design has the class ‘AbstractPart’ with two subclasses (Cpu and Gpu), they inherit the attributes and methods from the ‘AbstractPart’. The subclasses has polymorphism of the abstract method get_type() and get_description() , which means they redefined the same function in the subclass. The parameters entered to create the class objects are encapsulated and needs to be accessed through public method.

Why are your entity classes good abstractions of the real-world entities?

Because it covers most of the common attributes that would be needed for a retail environment, which may be expended to fit all kinds of computer components. It may not reflect all of the attributes of the real-world entities, but it is a good abstraction based on its use case.