- How does your design implement the four pillars of OOP (abstraction, encapsulation, inheritance and composition, and polymorphism)?
    - Abstraction: we have two abstract methods of **get_book_type** and **display_info**
    - Encapsulation**:** as we have private attributes, they are encapsulated in such a way that they cannot be called or accessed directly. By creating objects, we are encapsulating.
    - Inheritance & composition: ebook and textbook classes have inherited common attributes and methods of book class.
    - Library_stat object is an example of composition which means that library stats object is created in LibraryManager class.
    - Polymorphism: two methods of **get_book_type** and **display_info** are examples of abstract methods which we also use them in ebook and textbook classes. They behave differently based on the class they are in.

- Why are your classes good abstractions (i.e., models) of the real-world entities they represent?

Because they have real attributes and methods of real entity (library) thus they can behave as a real-world entity. In another world, they are encapsulated in such a way that only the needed and relevant details of the entity is shown to public interface.

eBook		
ID	**Method**	**Test Description**
01A	__init__	Valid **platform** and **book_category** and the book type is **ebook** → Creates a **ebook** object
01B	__init__	platform not in the list book_category not in the list Invalid type of platform value(!=str) and book_category(!= str)
02A	get_book_type	Gets the valid/not none book type (ebook) value from the object in setUp method
03A	get_book_platform	Gets the valid/not none book type (ebook) value from the object in setUp method
04A	get_available_platforms	Gets the valid/not none book platform value from the object in setUp method
05A	get_book_genre	Gets the valid/not none book genre (ebook) value from the object in setUp method
06A	suffix	Gets the valid ebook edition suffix from the object in setUp method
07A	display_info	Gets the valid/not none ebook availability status in the library from the object in setUp method

Textbook		
ID	**Method**	**Test Description**
01A	__init__	Valid **cover_type**, **book_subject** and the type is **textbook** → Creates a **textbook** object
01B	__init__	**cover_type** and **book_subject** not in the list specified Invalid type of cover type (!=str) and book subject (!)
02A	get_book_type	Gets the valid/not none book type value from the object in setUp method
03A	get_book_subject	Gets the valid/not none book subject value from the object in setUp method
04A	get_cover_type	Gets the valid/not none book cover type value from the object in setUp method
05A	get_available_cover_types	Gets the valid/not none book cover types available in the library value from the object in setUp method
06A	suffix	Gets the valid textbook edition suffix from the object in setUp method
07A	display_information	Gets the valid/not none textbook availability status in the library from the object in setUp method