```python
# StudentClass.py


class Student:

    count = 0

    def __init__(self, name=None,age = None,course = None, hometown=None):

        self.Name = name
        self.Age = age
        self.__Course = course
        self.__ID = None
        self._Hometown = hometown

        if name and age and course and hometown:
            self.__register()

    def get_name(self):
        return self.Name

    def get_age(self):
        return self.Age

    def get_course(self):
        return self.__Course

    def get_ID(self):
        return self.__ID

    def get_hometown(self):
        return self._Hometown

    # private method (call inside class only, not from outer side)
    def __register(self):
        Student.count += 1
        self.__ID = Student.count
        return self.__ID

```

```python
# LibraryItemCLass.py

from abc import ABC,abstractmethod

# Parent Class - abstract method
class LibraryItem(ABC):

    # constructor
    def __init__(self, count,id):
        self.count = count
        self.id = id

    def display_info(self):
        print(f"Count:{self.count}")
        print(f"id:{self.id}")

    @abstractmethod
    def check_out(self):
        print(f"Invoking absract method from parent class {__name__}")
```

```python
# BookClass.py

from LibraryItemClass import LibraryItem

# Child class
class Book(LibraryItem):
    def __init__(self, book_title, book_author,  book_category, book_id, count, id):
        super().__init__(count,id)

        # Private attributes
        self.__title = book_title
        self.__author = book_author
        self.__category = book_category
        self.__id = book_id

    # Method toprint book details
    def print_book_details(self):
        print(self.__title)
        print(self.__author)
        print(self.__category)
        print(self.__id)

    # Getter  for the title attribute
    def get_title(self):
        return self.__title

    def search(self,title,author=None):
        if author:
            print(f"searching book by author: {author} ")

        else:
            print(f"Searching by title: {title}")

    def check_out(self):
        print(f"Invoking absract method from child class {__name__}")
        # return super().check_out()
```