```
LA2main.py  # Main File

Import book_keeper file

main():
        control = an object from the controller class
        quit_flag = false  # When false the loop will continue to iterate
        while quit_flag is false:
                Display option to user
                Have user choose option 1, 2, or 3
                if option one:
                        show all of the library items
                elif option two:
                        Have user enter a call number
                        Print the the items information, including when it was checked out
                        and when it is due
                elif option three:
                        quit_flag = True
                else:
                        print ‘Invalid response’
```

book_keeper.py

```
import datetime, timedelta from datetime


class LibraryItem:  # Superclass
        __init__():
                Initializes the data for  call_number, checkout_type (book or periodical), the
                status of the item (checked_out), date_checked_out, and date_due

        check_out():
                checked_out = True
                if checkout_type is a book:
                        date_checked_out = today
                        date_due = today + 21
                elif checkout_type is a periodical:
                        date_checked_out = today
                        date_due = today + 7
                else:
                        checked_out = False  # Currently not another option available

        get_call_number():
                send back call_number to caller

        is_checked_out():
                sends back true or false boolean value of checked_out to caller

        get_date_checked_out():
                send back date_checked_out to caller

        get_date_due():
                send back date_due to caller

        set_date_due():
                sets date_due to new given value


class Book(LibraryItem)  # Sub-class
        __init__():
                Calls LibraryItem.__init__ to initialize checkout type and call_number
                Initializes the author, genre, and title itself
```

__str__():
    Creates a book string containing most data
    if the book is checked out:
        add that the book is checked out, checkout_day, and due_date onto the book string.
    else:
        Add that the book is not checked out onto the book string
    return book string

class Periodical(LibraryItem) # Sub-class
__init__():
    Calls LibraryItem.__init__ to initialize checkout type and call_number
    Initializes the volume, issue, subject, and title itself
__str__():
    Creates a periodical string containing most data
    if the periodical is checked out:
        Add that the periodical is checked out, checkout_day, and due_date onto the periodical string.
    else:
        Add that the periodical is not checked out onto the periodical string
    return book string

class Controller: # Controls how the data is displayed
__init__():
    Creates an empty list of library items. This will be used to store the library items in the input file

show_menu():
    Displays the menu options to the user.

display_collection():
    Displays the collection of library items on the screen

find_item(self, call_num):
    Searches in the list of library items for the item with the call number received as a parameter. # For loop
        if found before loop ends:
            return the item
    return 'nothing found'

check_out_materials(self, user_input)

    User enters a call_number as a parameter
    Call find_item to look for the item
    if item doesn't exist:
        print 'item was not found'
    else:
        check_out the item
        print(item)

read_input():

    Reads and organizes the data in the text file