

## Practical4

In this practical, a simple library program is written. It can achieve borrow, return and check available book operation. It consists of four ruby files.

**'Book.rb'** This is the class to store the details of book. A book object has six attributes, its title, book author, book id, publish date, pages and edition. Book id is a unique number and used as key to control books. There is a **'get\_id'** method that returns the book id.

**'user.rb'** This is the class to store the details of user. A user object has four attributes, user name, address, user id and borrow list. User id is a unique number. The borrow list stores the book id that the user has already borrowed. The **'get\_blist'** method returns the borrow list of the user. The **'borrow\_update'** method is used to update the borrow list of the user after a borrow operation by pushing the newly borrowed book id to the borrow list. The **'return\_update'** method is similar. It update the borrow list after a return operation by delete the book id from the borrow list.

**'library.rb'** This is the class to store details of library. A library object has two objects, **borrowed\_array** and **available\_array**. They are used to store which book is borrowed and which is available in the library. There are three methods in this class. The **'get\_available'** method returns the **available\_array** which is the available list of the library. The **'borrow\_update'** method is used to update the two arrays when borrow operation happen. It takes book id as input, delete this id in the **available\_array** and push the id into **borrowed\_array**. Another method **'return\_update'** is the opposite to the **'borrow\_update'** method. It controls the return operation, delete the id from **borrowed\_array** and push it into the **available\_array**.

**'Top.rb'** This is the top-level file. It has three main function, borrow book, return book and check available books. The **'borrow\_book'** method takes one user object, one book object and library object. It first check if the id of book in the available list of library using method in library class. If available, then call the update method in user class and library class to record that this user has borrow the book and this book is no longer available in the library. otherwise print a message telling the user this book is already borrowed by another user. The **'return\_book'** method also takes one user object, one book object and library object. It first get the borrow list form user and check if the user has borrowed this book. If not, the book can't be returned. A return operation also calls the update in user class and library class. The third function **'book\_available'** compare the available list in the library with the array that stores all the book id. If id matches, the book is still available and print out a message. Besides, in this file we create five book instances, two user instances and one library instance. Finally, I write some test cases to check the system functionalities. Besides some basic operations, some exceptions like borrow the same book twice or return some books that haven't borrowed to test the system robustness.