Test coverage criteria

While making the program:

Check each method.

If a method should have multiple paths, test each one.

If a method should throw an exception, test that path.

If a method should stop you from entering certain inputs, test that path

Goal is to isolate the cause of potential problems .

Book tests

* Test toString: compare to desired string
  + Test with no departments Constructor And with departments Constructor

* Set a books values to new ones and then get all the values and compare it to the expected
* Check that getDepartmentString returns all if no department is listed or the departments separated by “, “
* Test that taking out a loan on a book makes the book unavalible and returning it makes it available
* Check that inserting an invalid date throws an exeption

BookList test

* Test toString: compare to desired string
  + Test with no departments Constructor And with departments Constructor
* Test add and get
* Test that booklist equals expected after remove() book
* Check that getDepartmentString returns all if no department is listed or the departments separated by “, “

Ebook test

* Test constructor, book tests applies

LiberarySystem test

* Test get BookList gets desired books test with departments and without
* Test that getPerson throws error when the input id does not match existing user
* Test that signin returns false if wrong password is entered
* Test getSignedIn returns the person that was gotten if signIn was sucesssfull
* Test dateHander returns 1 if date1 is bigger or equal date2 (test day,month,year indiviually)
* Test dateHander returns 0 if date2 is bigger or equal date21(test day,month,year indiviually)
* Test dateHander returns -1 if date2 if input invalid
* Test that getBookOps
  + returns the “1)loan book” if book is available and is eligible and book is in department it and book is not ebook
  + returns the “1book is unavailable until[date] 1)reseve book” if book is unavalible and is eligible and book is in department it and book is not ebook
  + Returns “you must return [current loan] to take another” if book is available and sign in iis not eligible and book is in department it and book is not ebook
  + If book is in signed in is in department and is ebook return “1)read book, 2)download
  + If book in not in signedin s department  “return this book is not in your department”
* processBook test
  + Test that the process new book(test ebook and book, and test with and without departments) adds to the book to the booklist by comparing the last book in list toString to expected
  + Assert that it returns “not a valid date” if invalid date is put the date feild

Loan test

* Make loan
* Compare strings to expected strings for returned late books, unreturned late books, books still out on loan, and books returned normally date

Person test

* Test isStaff()
* Add loans to loans list and compare the loans to a test list of expected result
* Tes toString august expected for staff and student
* Test invalid email inputs throws RuntimeExeption
* Test adding a loan belonging to another person
* Test get and set name
* Test get and set departments

End to end Test:

Test each feature listed in user stories

Test convenience functions (Such as return to home)

Test coverage

The menu and simulator can't be tested with jira this drives down total line coverage a lot, but we should target a coverage around 80%, accepting classes with alot of getters and setters