# **MIS 301 – Programming Assignment 3**

**Due: Dec 5, 11:59 pm**

**Important Instructions:**

1. This is an individual assignment.
2. Please read the submission instruction below before you start to program.
3. You may discuss programming strategies with others, but you cannot share code or use code provided by others/other sources.
4. Assignments submitted after the deadline will automatically have a one-point deduction for each hour they are late.
5. You need to complete the assignments using only the Java classes covered in class.

**Submission Instructions:**

1. When setting up a project in Eclipse for this assignment, name your project **<Your Last Name>\_<First Name>\_S<Your Section Number>\_HW3**. For example, "Yin\_Junming\_S1\_HW3". Create a separate class (source code) for each question. The recommended names are Question1, Question2, and Question3. If you do that, your "src" folder should contain the following files: Question1.java, Question2.java, and Question3.java.
2. Once you finish programming, export the finished project as a zip file. In Eclipse, 1) File -> Export; 2) Select General > Archive File; 3) Select the project to be exported and choose the "Save in zip format" option; 4) Click Browse to choose the destination and the name of archive file; 5) Click Finish. You will find the zip file of your project at the chosen destination. Rename it using the same naming convention as your Eclipse project, e.g., **Yin\_Junming\_S1\_HW3.zip**.
3. For each question, copy and paste your source code from Eclipse into the box provided.
4. For each question, copy and paste a sample of your output from Eclipse into the box provided.
5. Submit both the Word document and the zip file on Blackboard. For the Word document, please rename it using the same naming convention as your Eclipse project, e.g., **Yin\_Junming\_S1\_HW3.docx**.

This assignment involves the development of a library system. The library can store **up to** 100 books. Each book has information of title, author, year, and the borrowed status (true or false). The library allows adding new books to it, checking out a book, and returning a book. We assume different books have different titles in the library.

The general rules of the systems include

* When adding a new book to the library, the borrowed status of that book is **false** by default. Make sure that you only add a new book to the library if the maximum capacity of library (100) has not been reached. If the maximum capacity is reached, no need to increase its maximum capacity and simply print an appropriate error message.
* If checking out a book that doesn’t belong to the library, print an appropriate error message. If checking out a book that does belong to the library but has been checked out, print an appropriate message. Otherwise, successfully check out that book and set its borrowed status to **true**.
* If returning a book that doesn’t belong to the library, print an appropriate error message. Otherwise, successfully return that book and set its borrowed status to **false**.

You need three classes:

1. Book: instance variables, constructors, getters and setters, and toString() method.
2. Library: instance variables, constructors, add(title, author, year), findBook(searchTitle), checkOutBook(title), returnBook(title), and toString() method.
3. MyLibrary: we already provide this driver class to you.

**Hint:**

* When implementing the method findBook in the Library class: if the specified book title is in the library, return its index in the array of Book references; otherwise, return a negative value such as -1 to indicate that the specified title is not in the library.
* You may find the slides of Class 19 and Class 20 as well as pp. 396-401 of the textbook helpful.
* If you have correct implementation, you should expect to the **exact output** at the end of this document.

**//Book.java**

**public** **class** Book {

//instance variables

**protected** String title; //using protected variables only accessible by these 3 classes

**protected** String author;

**protected** **int** year;

**protected** **boolean** borrowedStatus; //boolean for true or false

//Constructor, setters, and getters

**public** Book(String t, String a, **int** y, **boolean** borrStat){

title = t;

author = a;

year = y;

borrowedStatus = borrStat;

}

**public** String getTitleName() //returns title string

{

**return** title;

}

**public** String getAuthor() //returns author string

{

**return** author;

}

**public** **void** setYearPublish(**int** y)

{

year = y;

}

**public** **int** getYearPublish() //returns year int

{

**return** year;

}

**public** **boolean** getBorrowed()

{

**return** borrowedStatus; //returns boolean, false by default

}

**public** **void** setBorrowed(**boolean** borrStat)

{

borrowedStatus = borrStat;

}

**public** String toString()

{

String result;

//result = year + " " + title + " " + author + borrowedStatus;

result = String.*format*("%-6s", year) + String.*format*("%-26s", title) + String.*format*("%-20s", author);

**if**(borrowedStatus == **true**){

result += " borrowed";

}

**return** result;

}

}

**// Library.java**

**public class Library {**

**// instance variables**

**protected** **int** cap, count;

**protected** Book[] books;

**// constructor**

**public** Library ()

{

books = **new** Book[100];

count = 0;

}

**// add a book**

**public void add(String title, String author, int year){**

**if** (count < 100){ //only add a book if there is a room in the library

Book b = **new** Book(title, author, year, **false**);

books[count] = b;

count++; //adding one more book to the library

}

**else**{

System.***out***.println("Maximum capacity for library has already been reached!"); //error message when there's already 100 books in the library

}

**}**

**// find a book**

**public int findBook(String searchTitle) {**

**for** (**int** i = 0; i < count; i++) {

**if** (books[i].getTitleName().equals(searchTitle)){ //goes through all the books and compares titles

**return** i;

}

}

**return** -1; //if not in library returns negative number

**}**

**// check out a book**

**public void checkOutBook(String title) {**

**int** index;

index = findBook(title);

**if**(index == -1){

System.***out***.println(title + " does not belong to the library!");

}

**else** **if**(books[index].getBorrowed()== **true**) {

System.***out***.println(title + " has already been checked out!");

}

**else**{

books[index].setBorrowed(**true**);

System.***out***.println("You have successfully checked out " + title + "!");

}

**}**

**// return a book**

**public void returnBook(String title) {**

**int** index;

index = findBook(title); //makes sure book belongs in library

**if**(index == -1){

System.***out***.println(title + " does not belong in the library!");

}

**else**{

books[index].setBorrowed(**false**);

System.***out***.println("You have successfully returned " + title + "!"); //the book is in the library

}

**}**

**// toString method**

String result;

result = "------------------------------------\n";

result += "Number of books: " + count + "\n\n";

result += "Book List: \n";

**for** (**int** i = 0; i < count; i++){

result += books[i].toString()+"\n";

}

**return** result;

}

**}**

**// MyLibrary.java**

**public class MyLibrary {**

**public static void main(String[] args) {**

**Library myLibrary = new Library();**

**myLibrary.add("Investment Science", "David Luenberger", 2013);**

**myLibrary.add("Financial Modeling", "Simon Benninga", 2014);**

**myLibrary.add("Investment Banking", "Joshua Rosenbaum", 2013);**

**myLibrary.add("Neural Network Learning", "Martin Anthony", 2009);**

**myLibrary.add("Convex Optimization", "Steve Boyd", 2004);**

**myLibrary.add("Statistical Learning", "Gareth James", 2006);**

**System.*out*.println(myLibrary);**

**// check out two books**

**myLibrary.checkOutBook("Statistical Learning");**

**myLibrary.checkOutBook("Financial Modeling");**

**System.*out*.println(myLibrary);**

**// check out two books**

**myLibrary.checkOutBook("Machine Learning");**

**myLibrary.checkOutBook("Statistical Learning");**

**System.*out*.println(myLibrary);**

**// return two books**

**myLibrary.returnBook("Machine Learning");**

**myLibrary.returnBook("Statistical Learning");**

**System.*out*.println(myLibrary);**

**}**

**}**

**Output:**

|  |
| --- |
| ------------------------------------  Number of books: 6  Book List:  2013 Investment Science David Luenberger  2014 Financial Modeling Simon Benninga  2013 Investment Banking Joshua Rosenbaum  2009 Neural Network Learning Martin Anthony  2004 Convex Optimization Steve Boyd  2006 Statistical Learning Gareth James  You have successfully checked out Statistical Learning!  You have successfully checked out Financial Modeling!  ------------------------------------  Number of books: 6  Book List:  2013 Investment Science David Luenberger  2014 Financial Modeling Simon Benninga borrowed  2013 Investment Banking Joshua Rosenbaum  2009 Neural Network Learning Martin Anthony  2004 Convex Optimization Steve Boyd  2006 Statistical Learning Gareth James borrowed  Machine Learning does not belong to the library!  Statistical Learning has already been checked out!  ------------------------------------  Number of books: 6  Book List:  2013 Investment Science David Luenberger  2014 Financial Modeling Simon Benninga borrowed  2013 Investment Banking Joshua Rosenbaum  2009 Neural Network Learning Martin Anthony  2004 Convex Optimization Steve Boyd  2006 Statistical Learning Gareth James borrowed  Machine Learning does not belong to the library!  You have successfully returned Statistical Learning!  ------------------------------------  Number of books: 6  Book List:  2013 Investment Science David Luenberger  2014 Financial Modeling Simon Benninga borrowed  2013 Investment Banking Joshua Rosenbaum  2009 Neural Network Learning Martin Anthony  2004 Convex Optimization Steve Boyd  2006 Statistical Learning Gareth James |