

# Book Bank System

## Assignment 5

CPCS203 Programming-II Spring 2019

Assigned Date: Sunday 31/03/2019

**Delivery Date: Sunday 07/04/2019**



### Instructions

- This program must ONLY be submitted on the Blackboard!
- This project worth 6% of the overall module marks (100%).
- NO assignment will be accepted after 11:59 pm for any reason
- Students can submit their assignment between 11 and 11:59 PM but in this case, it will be considered as late submission, and they will lose 2 points from the total mark of the assignment.
- For discussion schedule, check the captain name, date and time on the BlackBoard.
- Further information is provided in the course syllabus.
- ***Further information is provided in the course syllabus.***

## Objectives

- Applying the Interface and Exception concept.
- Implementing Exception handling concepts (the try catch).
- Learn how to use and implement a simple GUI.

## How to submit your assignment?

- Submit your assignment on the Blackboard ONLY.
- Make sure to add your names / IDs / Section / Your name / Assignment number at the beginning of your program

## Files provided with assignment

- One input file: Input.txt
- Two output files: Output.txt, report.txt

## Description

Book Bank System is a simple system that allows students and faculty staff to borrow books, tapes and articles.

It must enable students and staff members to borrow books, tapes, or articles and allow them to borrow these materials only for a specified number of days. Then, a penalty fee will be issued if a borrower did not return an item within a specified duration.

This program contains a simple GUI that allows students and staff to check the amount of fee they should pay for late returns of materials by entering their ID.

### Classes and Interface details:

Classes	Comments
Materials.java	Given, DON'T MAKE ANY CHANGE IN THIS FILE
Article.java	Given, DON'T MAKE ANY CHANGE IN THIS FILE
Book.java	Given, DON'T MAKE ANY CHANGE IN THIS FILE
Tapes.java	Given, DON'T MAKE ANY CHANGE IN THIS FILE
Borrower.java	Given, Implement Payable interface
Student.java	Given, Complete the getFees() method
Staff.java	Given, Complete the getFees() method
Payable.java	Create this interface with a getFees() method that returns a double value.
MaterialNotFoundException.java	Create this sub class of Exception class
BookBank.java (GUI class)	Given, This class contains methods that needs to be implemented as per the given specification.

## The Initial Procedure of the Program

This program uses File I/O to read input from a given input file [input.txt].

### The commands you will have to implement are as follows:

**Add\_Borrower\_Student-** your program must read borrower information to be added to the system. The command will be followed by the following information, ALL on the same line:

an **int** representing **ID** of a student; a **String** representing **name** of a student; a **char** representing the **gender** of a student; an **int** representing **phone number** of a student; a **String** representing **major** of a student; a **String** representing the **title** of the **material** that is borrowed by the student; an **int** representing **borrowing duration in days (how many days he/she is allowed to borrow the material)**; [see input.txt]

Command Example
Add_Borrower_Student 1028666 Ahmad_Ali m 054579994 CS Aerodynamics_for_Engineering_Students 5

**Note: You have to check which material student is borrowing [Book , Tape , Article]**

**Add\_Borrower\_Staff\_member**—your program must read borrower information of the staff to be added to the system. The command will be followed by the following information, ALL on the same line:

an **int** representing **ID** of a staff; a **String** representing **name** of a staff; a **char** representing the **gender** of a staff; an **int** representing **phone** of a staff; a **String** representing **job** of a staff; a **String** representing the **title** of the **material** that is borrowed by the staff; an **int** representing **borrowing duration in days (how many days he/she is allowed to borrow the material)**; [see input.txt]

Command Example
Add_Borrower_Staff_member 1276187 Ziad_Malek m 054672819 TA Animal_Behavior 3

**Note: You have to check which material staff is borrowing [Book , Tape , Article]**

**Check\_Item\_Status**– Your program must check if an item is available or not. The command will be followed by the title of the item wanted to be checked, ALL on the same line:

A **String** representing the **title** of a material (book/tape/article); assuming there is no repeated items in our book bank. [see input.txt]

Command Example
Check_Item_Status Aerodynamics_for_Engineering_Students

**Note: Item may be [Book , Tape , Article]**

**Return\_Item**– This command will enable a student or a staff member to return a borrowed item to the book bank. The command will be followed by the following information, ALL on the same line:

an **int** representing the **ID** of the student or a staff member; a **String** representing the **title** of the **material** ; an **int** representing the **actual borrowing duration in days (for how many days they actually borrowed the material)**; [see input.txt]

Command Example
Return_Item 1000660 From_the_Oscillating_Universe_to_Relativistic_Energy 40

**Display\_Fees**– Your program must generate a report of required fees and store the output to an output file named **Report.txt**. [see Report.txt]

If the student does not return the material during the duration, he is charged **5 Riyals** late **fee** for each day after the deadline.

If the **staff member** does not return the material during the duration, he is charged **7 Riyals** late **fee** for each day after the deadline.

**for example**, let's say that they are allowed to borrow a specific book for 5 days but they return it after 7 days, in that case they need to pay for the extra 2 days.

### Important program details:

- Add a new **Custom Exception class** and name it **MaterialNotFoundException** this class is a direct subclass of the Exception class, it should specify the default message "Error: The material titled *material title* was not found in this Book Bank".
- If the program tries to look for a material that does not exist in the BookBank an object of the **MaterialNotFoundException** class is thrown.
- **Hint:** you will need to throw an exception when processing the following commands **Check\_Item\_Status, Add\_Borrower\_Student Add\_Borrower\_Staff\_member**.
- Use try-catch block to print the exception message to the file. [see output.txt]

## Program Graphical User Interface

The screenshot shows a window titled "Book Bank Management System". Inside, there's a "Welcome to Book Bank System" message. Below it, there are two main sections: "Borrower Information" and "Fee Information".

**Borrower Information:**

- Borrower ID:** A text box containing "1000666".

**Buttons:** Below the Borrower Information section, there are three buttons: "Search", "Clear", and "Exit".

**Fee Information:**

- Borrower Name:** A text box containing "Ahmad\_Adnan".
- Required Fee:** A text box containing "10.0".

Figure 1

This screenshot shows the same "Book Bank Management System" window, but with a "Warning!" dialog box overlaid in the center. The dialog box has a red octagonal icon with a white exclamation mark and the text "Borrower Not Found!". There is an "OK" button at the bottom right of the dialog box.

**Borrower Information:**

- Borrower ID:** A text box containing "23223".

**Buttons:** The "Search" button is visible below the Borrower ID field.

**Fee Information:**

- Borrower Name:** An empty text box.
- Required Fee:** An empty text box.

Figure 2

Implement your program using GUI which looks like the above figures. Allow the user (Staff or Student) to enter an ID number.

**Search button:** when press this button based on the entered borrower ID, an action will be triggered to search for the given ID in the file and return the required fees to be displayed on the form shown as per Figure 1.

Borrower ID should be entered using **TextField** then a Search button will be pressed. as a result the borrower name and required fee should be displayed using **TextField**.

In case if the Borrower ID doesn't exist, display Error message as per figure2.

**Clear button:** when press this button an action should be triggered to clear all entered information in the GUI-interface.

**Exit button:** implement the appropriate action to end the program.

### Important Notes:

- Use Interface, Exception handling concept to solve this assignment.
- Your program should output in a similar format to the sample run provided.
- Make your code readable.
- Document your code with comments.
- Use dash lines between each method.
- (Delayed submission will not be accepted).

### Deliverables:

You should submit one zip file containing all java codes:

BA\_1810348\_P5\_BookBankSystem where BA is your section, 1810348 your ID

**NOTE:** your name, ID, and section number should be included as comments in all files!

**Input and Output Format:** Your program should output in a similar format to the sample output file provided.

**Sample input:** See sample input file.

**Sample output:** See sample output files.

**Good Luck and Start Early!**