TAYLOR'S UNIVERSITY

Wisdom · Integrity · Excellence

TAYLOR'S UWE DUAL AWARDS PROGRAMMES MARCH 2021 SEMESTER

Object-oriented programming (ITS63304)

Final assessment

STUDENT DECLARATION:

1. **I confirm that I am aware of the University's Regulation Governing Cheating in a University Test and Assignment and of the guidance issued by the School of Computing and Engineering (SCE) concerning plagiarism and proper academic practice, and that the assessed work now submitted is in accordance with this regulation and guidance.**
2. **I understand that, unless already agreed with the School of Computing and Engineering (SCE), assessed work may not be submitted that hAS previously been submitted, either in whole or in part, at this or any other institution.**

Student Name	Student ID	Date	Signature	Score
Lim Kang Wei	0347555	8 July 2021		

Table of Content

Contents

Section A

1. Design the objects, its features and required behaviours for the given system. You need to provide justification for your design. (5 marks)

a) User

```
String username;
String usertype;
String userpassword;
int charged_amount;
ArrayList<String> borrowed = new ArrayList<>();
ArrayList<String> borrowing = new ArrayList<>();
int studentbookcountlimit = 5;
int staffbookcountlimit = 10;
```

Store:

i) Username, user type, user password
ii) User borrow history
iii) User current book borrowed
iv) User book count limit (student=5, staff=10)
v) User charged amount(for overdue)

b) Book(Library)

```
public String title;
public String author;
public String category;
public String synopsis = "No one has write a synopsis yet.";
public String status = "available";
public String overduedate = "-";
public String takenby;
public String takenuntil;
public int days_till_overdue;
public String reservedby;
public int numberborrowed;
```

Store:

i) Book title, author, category, synopsis
ii) Borrowed by and reserved by whom
iii) Borrowed until when
iv) Times they are borrowed
v) Borrow duration(student=30 days, staff=90 days)

2. Describe where and how you can use encapsulation to secure your code. You need to justify your answer. (3 marks)

```
if (Fatimah_logged_in || Ahmad_logged_in || Damian_logged_in) {

    System.out.println("How can we help you?");
    System.out.println("1. View book's author, title, and category");
    System.out.println("2. View book status");
    System.out.println("3. View book synopsis");

    if (is_librarian) {
        System.out.println("4. View member's details.");
        System.out.println("5. Check the person's charged amount.");
        System.out.println("6. Collect overdue charges.");
        System.out.println("7. View total overdue charges received.");
        System.out.println("8. Take the book back due to overdue.");
        System.out.println("9. Write synopsis for the book.");
        System.out.println("10. Change member's username.");
        System.out.println("11. Change member's password.");
        System.out.println("12. View that member's currently borrowed book.");
        System.out.println("13. View that member's borrow history.");
        System.out.println("14. Check how many times has the book been borrrowed.");
        System.out.println("15. View total for each book status.");
        System.out.println("16. Exit program.");
```

```
else if (is_member) {
  System.out.println("4. View available books.");
  System.out.println("5. Borrow book(if available).");
  System.out.println("6. Renew book.");
  System.out.println("7. Return book.");
  System.out.println("8. Reserve book.");
  System.out.println("9. View remaining days until the book overdues.");
  System.out.println("10. View how many more books can borrow.");
  System.out.println("11. View personal details.");
  System.out.println("12. Exit program.");
```

I used if-else statement at the part where user log in. If they are student, or staff, or librarian, their access of methods will be different.

In order to protect my data, especially when creating them, I will use getters and setters to retrieve datas related to books and users. Below are just a few of the methods.

```
public void getsynopsis() {
    System.out.println("Synopsis: " + this.synopsis);
}

public void getdetails() {
    System.out.println("Title: " + gettitle());
    System.out.println("Author: " + getauthor());
    System.out.println("Category: " + getcategory());
    System.out.println("Status: " + status);
}

public void settitle(String title) {
    this.title = title;
}

public void setauthor(String author) {
    this.author = author;
}
```

```
public void getuserdetails() {
    System.out.println("Username: " + getusername());
    System.out.println("User type: " + getusertype());
    System.out.println("Password: " + getuserpassword());
    System.out.println("Book(s) borrowed: " + borrowed);
    System.out.println("Total charges: " + charged_amount);
}

public void addcharges(int amount) {
    this.charged_amount += amount;
}

public void getcharges() {
    System.out.println("This person has been charged RM" + charged_amount + " till date.");
}

public int grabcharges() {
    return charged_amount;
}

public void addbook(String book) {
    borrowed.add(book);
}
```

I used multiple boolean statements to make sure the username, user type and password given by the user match the record. Or else the user won't be able to proceed.

```
Boolean Ahmad_logged_in = user.equals("Ahmad") && pass.equals("Apple") && acctype.equals("student");
Boolean Damian_logged_in = user.equals("Damian") && pass.equals("Durian") && acctype.equals("staff");
Boolean Fatimah_logged_in = user.equals("Fatimah") && pass.equals("Blueberry") && acctype.equals("librarian");

Boolean is_librarian = user.equals("Fatimah") && acctype.equals("librarian");
Boolean is_member = (user.equals("Ahmad") && acctype.equals("student")) || (user.equals("Damian") && acctype.equals("staff"));

Boolean is_book1 = book.equals("Naruto") || book.equals("Masashi Kishimoto");
Boolean is_book2 = book.equals("Lord of the Rings") || book.equals("J. R. R. Tolkien");
Boolean is_book3 = book.equals("Attack on Titan") || book.equals("Hajime Isayama");

Boolean is_Ahmad = person.equals("Ahmad");
Boolean is_Damian = person.equals("Damian");
```

```
run:
Which account do you want to log in?
Ahmad
What's the type of your account?
student
Key in password.
Orange
Which book do you want to interact with?
Naruto
Which person do you want to interact with?
none
Error
```

3. Describe where and how you can use exception to minimize possible bug in your code. You need to justify your answer (3 marks)

I threw ParseExceptions at the part where the program calculates and verify the duration of borrowing and renewing a book. Because that part involves lots of String-LocalDate conversion(parsing). Below show one of the two exceptions(one for student, another for staff).

```
public void setstudtakenuntil(String date) throws ParseException {
    if (date.equals("-")) {
        days_till_overdue = 0;
    } else {
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("d/MM/yyyy");
        LocalDate a = LocalDate.now();
        String b = date;
        LocalDate c = LocalDate.parse(b, formatter);
        long result = ChronoUnit.DAYS.between(a, c);
        days_till_overdue = (int) result;
        if (days_till_overdue < 30) {
            days_till_overdue = (int) result;
        } else {
            System.out.println("Exceeded 30 days. \n System will shut down.");
            System.exit(0);
        }
    }
}
```

4. Suggest two features that are unique and important in this system. Elaborate the feature in detail. (4 marks)

   First feature:

   User is not allowed to renew overdue date of the book if they didn't borrow the book at first.

```
run:
Which account do you want to log in?
Ahmad
What's the type of your account?
student
Key in password.
Apple
Which book do you want to interact with?
Naruto
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
6
Error
```

Second feature:

When borrow or renew, my app will check whether the date inserted exceeds the duration limit.(Staff=90 days, student=10 days)

```
public void setstudtakenuntil(String date) throws ParseException {
    if (date.equals("-")) {
        days_till_overdue = 0;
    } else {
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("d/MM/yyyy");
        LocalDate a = LocalDate.now();
        String b = date;
        LocalDate c = LocalDate.parse(b, formatter);
        long result = ChronoUnit.DAYS.between(a, c);
        days_till_overdue = (int) result;
        if (days_till_overdue < 30) {
            days_till_overdue = (int) result;
        } else {
            System.out.println("Exceeded 30 days. \n System will shut down.");
            System.exit(0);
        }
    }
}
public void setstaftakenuntil(String date) throws ParseException {
    if (date.equals("-")) {
        days_till_overdue = 0;
    } else {
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("d/MM/yyyy");
        LocalDate a = LocalDate.now();
        String b = date;
        LocalDate c = LocalDate.parse(b, formatter);
        long result = ChronoUnit.DAYS.between(a, c);
        days_till_overdue = (int) result;
        if (days_till_overdue < 90) {
            days_till_overdue = (int) result;
        } else {
            System.out.println("Exceeded 90 days. \n System will shut down.");
            System.exit(0);
        }
    }
}
```

```
run:
Which account do you want to log in?
Damian
What's the type of your account?
staff
Key in password.
Durian
Which book do you want to interact with?
Naruto
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
5
You can borrow this book for 90 days. (Type in when you want to return the book in this format: dd/MM/yyyy)
08/07/2022
Exceeded 90 days.
 System will shut down.
BUILD SUCCESSFUL (total time: 31 seconds)
```

```
run:
Which account do you want to log in?
Ahmad
What's the type of your account?
student
Key in password.
Apple
Which book do you want to interact with?
Lord of the Rings
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
5
You can only borrow this book for 30 days. (Type in when you want to return the book in this format: dd/MM/yyyy)
08/07/2022
Exceeded 30 days.
 System will shut down.
BUILD SUCCESSFUL (total time: 23 seconds)
```

Section B

1. The program can be logged in by librarian and member (student and staff).

```
run:
Which account do you want to log in?
Ahmad
What's the type of your account?
student
Key in password.
Apple
Which book do you want to interact with?
Naruto
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
```

```
run:
Which account do you want to log in?
Damian
What's the type of your account?
staff
Key in password.
Durian
Which book do you want to interact with?
Lord of the Rings
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
```

```
run:
Which account do you want to log in?
Fatimah
What's the type of your account?
librarian
Key in password.
Blueberry
Which book do you want to interact with?
none
Which person do you want to interact with?
Ahmad
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View member's details.
5. Check the person's charged amount.
6. Collect overdue charges.
7. View total overdue charges received.
8. Take the book back due to overdue.
9. Write synopsis for the book.
10. Change member's username.
11. Change member's password.
12. View that member's currently borrowed book.
13. View that member's borrow history.
14. Check how many times has the book been borrrowed.
15. View total for each book status.
16. Exit program.
```

2.

a) Search and view for book by author, title, and category.

```
run:
Which account do you want to log in?
Ahmad
What's the type of your account?
student
Key in password.
Apple
Which book do you want to interact with?
Naruto
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
1
Title: Naruto
Author: Masashi Kishimoto
Category: Manga
Status: available
```

```
run:
Which account do you want to log in?
Fatimah
What's the type of your account?
librarian
Key in password.
Blueberry
Which book do you want to interact with?
Attack on Titan
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View member's details.
5. Check the person's charged amount.
6. Collect overdue charges.
7. View total overdue charges received.
8. Take the book back due to overdue.
9. Write synopsis for the book.
10. Change member's username.
11. Change member's password.
12. View that member's currently borrowed book.
13. View that member's borrow history.
14. Check how many times has the book been borrrowed.
15. View total for each book status.
16. Exit program.
1
Title: Attack on Titan
Author: Hajime Isayama
Category: Manga
Status: available
```

b) View book status (available, borrowed, and reserved).

```
run:
Which account do you want to log in?
Damian
What's the type of your account?
staff
Key in password.
Durian
Which book do you want to interact with?
Lord of the Rings
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
2
Status: available
```

```
run:
Which account do you want to log in?
Fatimah
What's the type of your account?
librarian
Key in password.
Blueberry
Which book do you want to interact with?
Attack on Titan
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View member's details.
5. Check the person's charged amount.
6. Collect overdue charges.
7. View total overdue charges received.
8. Take the book back due to overdue.
9. Write synopsis for the book.
10. Change member's username.
11. Change member's password.
12. View that member's currently borrowed book.
13. View that member's borrow history.
14. Check how many times has the book been borrrowed.
15. View total for each book status.
16. Exit program.
2
Status: available
```

c) View book synopsis

Only librarian can write a book's synopsis.

```
run:
Which account do you want to log in?
Ahmad
What's the type of your account?
student
Key in password.
Apple
Which book do you want to interact with?
Naruto
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
3
Synopsis: No one has write a synopsis yet.
```

3.

a) Manage member’s details.

```
run:
Which account do you want to log in?
Fatimah
What's the type of your account?
librarian
Key in password.
Blueberry
Which book do you want to interact with?
none
Which person do you want to interact with?
Ahmad
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View member’s details.
5. Check the person's charged amount.
6. Collect overdue charges.
7. View total overdue charges received.
8. Take the book back due to overdue.
9. Write synopsis for the book.
10. Change member's username.
11. Change member's password.
12. View that member's currently borrowed book.
13. View that member's borrow history.
14. Check how many times has the book been borrrowed.
15. View total for each book status.
16. Exit program.
4
Username: Ahmad
User type: student
Password: Apple
Book(s) borrowed: []
Total charges: 0
```

b) View total for each book status.

Connecting from the last task, my application allows user to proceed using the same info(same book and same person they want interact with) without having to log in again.

```
Do you wish to continue using this account?
yes
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View member’s details.
5. Check the person's charged amount.
6. Collect overdue charges.
7. View total overdue charges received.
8. Take the book back due to overdue.
9. Write synopsis for the book.
10. Change member's username.
11. Change member's password.
12. View that member's currently borrowed book.
13. View that member's borrow history.
14. Check how many times has the book been borrrowed.
15. View total for each book status.
16. Exit program.
15
Available book count: 3
Borrowed book count: 0
```

c) View member borrowing status

In this case, Ahmad borrows 2 books. And Fatimah is able to check which book is borrowed from him.

```
Which account do you want to log in?
Fatimah
What's the type of your account?
librarian
Key in password.
Blueberry
Which book do you want to interact with?
none
Which person do you want to interact with?
Ahmad
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View member’s details.
5. Check the person's charged amount.
6. Collect overdue charges.
7. View total overdue charges received.
8. Take the book back due to overdue.
9. Write synopsis for the book.
10. Change member's username.
11. Change member's password.
12. View that member's currently borrowed book.
13. View that member's borrow history.
14. Check how many times has the book been borrrowed.
15. View total for each book status.
16. Exit program.
12
Borrowing: [Naruto, Lord of the Rings]
```

d) View member late charges

```
run:
Which account do you want to log in?
Fatimah
What's the type of your account?
librarian
Key in password.
Blueberry
Which book do you want to interact with?
none
Which person do you want to interact with?
Ahmad
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View member’s details.
5. Check the person's charged amount.
6. Collect overdue charges.
7. View total overdue charges received.
8. Take the book back due to overdue.
9. Write synopsis for the book.
10. Change member's username.
11. Change member's password.
12. View that member's currently borrowed book.
13. View that member's borrow history.
14. Check how many times has the book been borrrowed.
15. View total for each book status.
16. Exit program.
5
This person has been charged RM0 till date.
```

d) Return overdue book and collect overdue charges.

In this case, the book that Ahmad borrowed has overdue, and Fatimah wants to take the book back n charge Ahmad RM10 for not returning the book back to the library on time.

```
Do you wish to continue using this account?
no
Which account do you want to log in?
Fatimah
What's the type of your account?
librarian
Key in password.
Blueberry
Which book do you want to interact with?
Naruto
Which person do you want to interact with?
Ahmad
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View member's details.
5. Check the person's charged amount.
6. Collect overdue charges.
7. View total overdue charges received.
8. Take the book back due to overdue.
9. Write synopsis for the book.
10. Change member's username.
11. Change member's password.
12. View that member's currently borrowed book.
13. View that member's borrow history.
14. Check how many times has the book been borrrowed.
15. View total for each book status.
16. Exit program.
8
```

```
Do you wish to continue using this account?
yes
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View member's details.
5. Check the person's charged amount.
6. Collect overdue charges.
7. View total overdue charges received.
8. Take the book back due to overdue.
9. Write synopsis for the book.
10. Change member's username.
11. Change member's password.
12. View that member's currently borrowed book.
13. View that member's borrow history.
14. Check how many times has the book been borrrowed.
15. View total for each book status.
16. Exit program.
6
How much are you collecting?
10
You charged this person RM10.
```

e) View member borrowing history

```
Do you wish to continue using this account?
no
Which account do you want to log in?
Fatimah
What's the type of your account?
librarian
Key in password.
Blueberry
Which book do you want to interact with?
none
Which person do you want to interact with?
Ahmad
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View member's details.
5. Check the person's charged amount.
6. Collect overdue charges.
7. View total overdue charges received.
8. Take the book back due to overdue.
9. Write synopsis for the book.
10. Change member's username.
11. Change member's password.
12. View that member's currently borrowed book.
13. View that member's borrow history.
14. Check how many times has the book been borrrowed.
15. View total for each book status.
16. Exit program.
13
Book(s) borrowed: [Naruto, Lord of the Rings]
```

In this case, Ahmad has borrowed 2 books, no matter whether has he returned the books, the history will not change.

f) View total overdue charges received

```
Do you wish to continue using this account?
yes
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View member’s details.
5. Check the person's charged amount.
6. Collect overdue charges.
7. View total overdue charges received.
8. Take the book back due to overdue.
9. Write synopsis for the book.
10. Change member's username.
11. Change member's password.
12. View that member's currently borrowed book.
13. View that member's borrow history.
14. Check how many times has the book been borrrowed.
15. View total for each book status.
16. Exit program.
7
Total overdue charges received is RM0.
```

In this case, no one has been charged yet. Hence it's zero.

4.

a) View personal detail.

```
run:
Which account do you want to log in?
Ahmad
What's the type of your account?
student
Key in password.
Apple
Which book do you want to interact with?
Naruto
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
11
Currently borrowing: []
Borrow history: []
Still can borrow: 10
Currently charged for: RM0
```

bi) Perform borrow, if book status is available.

In this case, Damian check whether Naruto is available in the library. Once the book's availability is confirmed, he performed borrow. Then Fatimah can check which book is Damian borrowing now.

```
run:
Which account do you want to log in?
Damian
What's the type of your account?
staff
Key in password.
Durian
Which book do you want to interact with?
Naruto
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
2
Status: available
```

1

```
Do you wish to continue using this account?
yes
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
5
You can borrow this book for 90 days. (Type in when you want to return the book in this format: dd/MM/yyyy)
10/07/2021
```

2

```
Do you wish to continue using this account?
no
Which account do you want to log in?
Fatimah
What's the type of your account?
librarian
Key in password.
Blueberry
Which book do you want to interact with?
none
Which person do you want to interact with?
Damian
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View member's details.
5. Check the person's charged amount.
6. Collect overdue charges.
7. View total overdue charges received.
8. Take the book back due to overdue.
9. Write synopsis for the book.
10. Change member's username.
11. Change member's password.
12. View that member's currently borrowed book.
13. View that member's borrow history.
14. Check how many times has the book been borrrowed.
15. View total for each book status.
16. Exit program.
12
Borrowing: [Naruto]
```

3

bii) Perform renew before overdue date, if borrowed by the user and not due.

In this case, Ahmad accidentally book the due date too early till date.

```
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
5
You can only borrow this book for 30 days. (Type in when you want to return the book in this format: dd/MM/yyyy)
10/07/2021
```

He wants to renew the book before the due date.

Then he is able to check the remaining days for him to read the book.

```
Do you wish to continue using this account?
yes
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
6
15/07/2021
```

```
Do you wish to continue using this account?
yes
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
9
7 days till overdue.
```

biii) Perform return, if borrowed by the user and not due.

Once Damian return the book, we can check what else is he borrowing now, also we can check his borrowing history.

1

```
run:
Which account do you want to log in?
Damian
What's the type of your account?
staff
Key in password.
Durian
Which book do you want to interact with?
Naruto
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
5
You can borrow this book for 90 days. (Type in when you want to return the book in this format: dd/MM/yyyy)
9/07/2021
```

2

```
Do you wish to continue using this account?
yes
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
7
```

3

```
Do you wish to continue using this account?
yes
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
11
Currently borrowing: []
Borrow history: [Naruto]
Still can borrow: 10
Currently charged for: RM0
```

5. Logout system to allow another user to login.

```
Do you wish to continue using this account?
no
Which account do you want to log in?
Ahmad
What's the type of your account?
student
Key in password.
Apple
Which book do you want to interact with?
Lord of the Rings
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
```

Connecting to the previous taks, we can decide whether to log out or continue using the same user. Even if we were to change the book or person we want to interact with, also need to log out and retype all new informations. The example on the left side is the user logging out from Damian's account and logging into Ahmad's account.

6. Exit system will close the program.

Refer to the previous picture, if we choose option #12. We will exit the entire program and shut down every record.

```
Do you wish to continue using this account?
no
Which account do you want to log in?
Ahmad
What's the type of your account?
student
Key in password.
Apple
Which book do you want to interact with?
Lord of the Rings
Which person do you want to interact with?
none
How can we help you?
1. View book's author, title, and category
2. View book status
3. View book synopsis
4. View available books.
5. Borrow book(if available).
6. Renew book.
7. Return book.
8. Reserve book.
9. View remaining days until the book overdues.
10. View how many more books can borrow.
11. View personal details.
12. Exit program.
12
Thank you for using our system.
BUILD SUCCESSFUL (total time: 7 minutes 51 seconds)
```

Appendix

Library_Management_java

```
package library_management_system;


import java.text.ParseException;
import java.util.ArrayList;
import java.util.Scanner;


public class Library_Management_System {


    public static void main(String[] args) throws ParseException {


        Scanner sc = new Scanner(System.in);
        int totalchargesgiven = 0;


        Library book1 = new Library("Naruto", "Masashi Kishimoto", "Manga");
        Library book2 = new Library("Lord of the Rings", "J. R. R. Tolkien", "Heroic romance");
        Library book3 = new Library("Attack on Titan", "Hajime Isayama", "Manga");


        User Ahmad = new User("Ahmad", "student", "Apple");
        User Damian = new User("Damian", "staff", "Durian");
        User Fatimah = new User("Fatimah", "librarian", "Blueberry");


        String answer;
```

```
do {
    System.out.println("Which account do you want to log in?");
    String user = sc.nextLine();
    System.out.println("What's the type of your account?");
    String acctype = sc.nextLine();
    System.out.println("Key in password.");
    String pass = sc.nextLine();
    System.out.println("Which book do you want to interact with?");
    String book = sc.nextLine();
    System.out.println("Which person do you want to interact with?");
    String person = sc.nextLine();

    Boolean Ahmad_logged_in = user.equals("Ahmad") && pass.equals("Apple") && acctype.equals("student");
    Boolean Damian_logged_in = user.equals("Damian") && pass.equals("Durian") && acctype.equals("staff");
    Boolean Fatimah_logged_in = user.equals("Fatimah") && pass.equals("Blueberry") && acctype.equals("librarian");

    Boolean is_librarian = user.equals("Fatimah") && acctype.equals("librarian");
    Boolean is_member = (user.equals("Ahmad") && acctype.equals("student")) || (user.equals("Damian") && acctype.equals("staff"));

    Boolean is_book1 = book.equals("Naruto") || book.equals("Masashi Kishimoto");
```

```
Boolean is_book2 = book.equals("Lord of the Rings") || book.equals("J. R. R. Tolkien");
Boolean is_book3 = book.equals("Attack on Titan") || book.equals("Hajime Isayama");

Boolean is_Ahmad = person.equals("Ahmad");
Boolean is_Damian = person.equals("Damian");

do {
   if (Fatimah_logged_in || Ahmad_logged_in || Damian_logged_in) {

      System.out.println("How can we help you?");
      System.out.println("1. View book's author, title, and category");
      System.out.println("2. View book status");
      System.out.println("3. View book synopsis");

      if (is_librarian) {
         System.out.println("4. View member’s details.");
         System.out.println("5. Check the person's charged amount.");
         System.out.println("6. Collect overdue charges.");
         System.out.println("7. View total overdue charges received.");
         System.out.println("8. Take the book back due to overdue.");
         System.out.println("9. Write synopsis for the book.");
         System.out.println("10. Change member's username.");
         System.out.println("11. Change member's password.");
         System.out.println("12. View that member's currently borrowed book.");
```

```
System.out.println("13. View that member's borrow history.");
System.out.println("14. Check how many times has the book been borrrowed.");
System.out.println("15. View total for each book status.");
System.out.println("16. Exit program.");

String Z = sc.nextLine();
switch (Z) {
    case "1" -> {
        if (is_book1) {
            book1.getdetails();
        } else if (is_book2) {
            book2.getdetails();
        } else if (is_book3) {
            book3.getdetails();
        }
        break;
    }

    case "2" -> {
        if (is_book1) {
            book1.getstatus();
        } else if (is_book2) {
            book2.getstatus();
        } else if (is_book3) {
```

```
        book3.getstatus();
    }
    break;
}

case "3" -> {
    if (is_book1) {
        book1.getsynopsis();
    } else if (is_book2) {
        book2.getsynopsis();
    } else if (is_book3) {
        book3.getsynopsis();
    }
    break;
}

case "4" -> {
    if (is_Ahmad) {
        Ahmad.getuserdetails();
    } else if (is_Damian) {
        Damian.getuserdetails();
    }
    break;
}
```

```
case "5" -> {
    if (is_Ahmad) {
        Ahmad.getcharges();
    } else if (is_Damian) {
        Damian.getcharges();
    }
    break;
}

case "6" -> {
    System.out.println("How much are you collecting?");
    int money = sc.nextInt();
    totalchargesgiven += money;
    if (is_Ahmad) {
        Ahmad.addcharges(money);
    } else if (is_Damian) {
        Damian.addcharges(money);
    }
    System.out.println("You charged this person RM" + money + ".");
    break;
}

case "7" -> {
```

```
                System.out.println("Total overdue charges received is RM" + totalchargesgiven + ".");
                break;
            }

            case "8" -> {
                if (is_book1) {
                    if (is_Ahmad) {
                        Ahmad.dropborrowing(book1.gettitle());
                        book1.setstudtakenuntil("-");
                    } else {
                        Damian.dropborrowing(book1.gettitle());
                        book1.setstaftakenuntil("-");
                    }
                    book1.settakenby("none");
                    book1.setstatus("available");
                    book1.setoverduedate("-");

                } else if (is_book2) {
                    if (is_Ahmad) {
                        Ahmad.dropborrowing(book2.gettitle());
                        book2.setstudtakenuntil("-");
                    } else {
                        Damian.dropborrowing(book2.gettitle());
                        book2.setstaftakenuntil("-");
```

```
            }
            book2.settakenby("none");
            book2.setstatus("available");
            book2.setoverduedate("-");
        } else if (is_book3) {
            if (is_Ahmad) {
                Ahmad.dropborrowing(book3.gettitle());
                book3.setstudtakenuntil("-");
            } else {
                Damian.dropborrowing(book3.gettitle());
                book3.setstaftakenuntil("-");
            }
            book3.settakenby("none");
            book3.setstatus("available");
            book3.setoverduedate("-");
        }
        break;
    }

    case "9" -> {
        if (is_book1) {
            book1.setsynopsis(sc.nextLine());
        } else if (is_book2) {
            book2.setsynopsis(sc.nextLine());
```

```
    } else if (is_book3) {
        book3.setsynopsis(sc.nextLine());
    }
    break;
}

case "10" -> {
    System.out.println("What's the new username?");
    if (is_Ahmad) {
        Ahmad.setusername(sc.nextLine());
    } else if (is_Damian) {
        Damian.setusername(sc.nextLine());
    }
    break;
}

case "11" -> {
    if (is_Ahmad) {
        Ahmad.setuserpassword(sc.nextLine());
    } else if (is_Damian) {
        Damian.setuserpassword(sc.nextLine());
    }
    break;
}
```

```
case "12" -> {
    if (is_Ahmad) {
        Ahmad.getcurrentbookborrowed();
    } else if (is_Damian) {
        Damian.getcurrentbookborrowed();
    }
    break;
}

case "13" -> {
    if (is_Ahmad) {
        Ahmad.gethistory();
    } else if (is_Damian) {
        Damian.gethistory();
    }
    break;
}

case "14" -> {
    if (is_book1) {
        book1.getborrownumber();
    } else if (is_book2) {
        book2.getborrownumber();
```

```
    } else if (is_book3) {
        book3.getborrownumber();
    }
    break;
}

case "15" -> {
    int availablecount = 0;
    if (book1.grabstatus().equals("available")) {
        availablecount++;
    }
    if (book2.grabstatus().equals("available")) {
        availablecount++;
    }
    if (book3.grabstatus().equals("available")) {
        availablecount++;
    }
    System.out.println("Available book count: " + availablecount);
    System.out.println("Borrowed book count: " + (3 - availablecount));
}
case "16" -> {
    System.out.println("Thank you for using our system.");
    System.exit(0);
    break;
```

```
        }
    }

} else if (is_member) {
    System.out.println("4. View available books.");
    System.out.println("5. Borrow book(if available).");
    System.out.println("6. Renew book.");
    System.out.println("7. Return book.");
    System.out.println("8. Reserve book.");
    System.out.println("9. View remaining days until the book overdues.");
    System.out.println("10. View how many more books can borrow.");
    System.out.println("11. View personal details.");
    System.out.println("12. Exit program.");

    String Z = sc.nextLine();
    switch (Z) {

        case "1" -> {
            if (is_book1) {
                book1.getdetails();
            } else if (is_book2) {
                book2.getdetails();
            } else if (is_book3) {
                book3.getdetails();
```

```
    }
    break;
}

case "2" -> {
    if (is_book1) {
        book1.getstatus();
    } else if (is_book2) {
        book2.getstatus();
    } else if (is_book3) {
        book3.getstatus();
    }
    break;
}

case "3" -> {
    if (is_book1) {
        book1.getsynopsis();
    } else if (is_book2) {
        book2.getsynopsis();
    } else if (is_book3) {
        book3.getsynopsis();
    }
    break;
```

```
}

case "4" -> {
    ArrayList<String> availist = new ArrayList<>();
    if (book1.grabstatus().equals("available")) {
        availist.add(book1.gettitle());
    }
    if (book2.grabstatus().equals("available")) {
        availist.add(book2.gettitle());
    }
    if (book3.grabstatus().equals("available")) {
        availist.add(book3.gettitle());
    }
    System.out.println(availist);
}

case "5" -> {
    if (is_book1 && book1.grabstatus().equals("available")) {
        book1.setstatus("borrowed");
        book1.setnewnumber();
        if (Ahmad_logged_in) {
            System.out.println("You can only borrow this book for 30 days. (Type in when you want to return the book in this format: dd/MM/yyyy)");
            book1.setstudtakenuntil(sc.nextLine());
            Ahmad.setborrowing(book1.gettitle());
```

```
                Ahmad.addbook(book1.gettitle());
                Ahmad.setstudentlimit(1);
            } else if (Damian_logged_in) {
                System.out.println("You can borrow this book for 90 days. (Type in when you want to return the book in this format: dd/MM/yyyy)");
                book1.setstaftakenuntil(sc.nextLine());
                Damian.setborrowing(book1.gettitle());
                Damian.addbook(book1.gettitle());
                Damian.setstafflimit(1);
            }

        } else if (is_book2 && book2.grabstatus().equals("available")) {
            book2.setstatus("borrowed");
            book2.setnewnumber();
            if (Ahmad_logged_in) {
                System.out.println("You can only borrow this book for 30 days. (Type in when you want to return the book in this format: dd/MM/yyyy)");
                book2.setstudtakenuntil(sc.nextLine());
                Ahmad.setborrowing(book2.gettitle());
                Ahmad.addbook(book2.gettitle());
                Ahmad.setstudentlimit(1);
            } else if (Damian_logged_in) {
                System.out.println("You can borrow this book for 90 days. (Type in when you want to return the book in this format: dd/MM/yyyy)");
                book2.setstaftakenuntil(sc.nextLine());
```

```
                Damian.setborrowing(book2.gettitle());
                Damian.addbook(book2.gettitle());
                Damian.setstafflimit(1);
            }
        } else if (is_book3 && book3.grabstatus().equals("available")) {
            book3.setstatus("borrowed");
            book3.setnewnumber();
            if (Ahmad_logged_in) {
                System.out.println("You can only borrow this book for 30 days. (Type in when you want to return the book in this format: dd/MM/yyyy)");
                book3.setstudtakenuntil(sc.nextLine());
                Ahmad.setborrowing(book3.gettitle());
                Ahmad.addbook(book3.gettitle());
                Ahmad.setstudentlimit(1);
            } else if (Damian_logged_in) {
                System.out.println("You can borrow this book for 90 days. (Type in when you want to return the book in this format: dd/MM/yyyy)");
                book3.setstaftakenuntil(sc.nextLine());
                Damian.setborrowing(book3.gettitle());
                Damian.addbook(book3.gettitle());
                Damian.setstafflimit(1);
            }
        } else {
            System.out.println("Book is taken.");
        }
```

```
    break;
}

case "6" -> {
    if (is_book1 && book1.grabstatus().equals("borrowed")) {
        if (Ahmad_logged_in) {
            book1.setstudtakenuntil(sc.nextLine());
        } else {
            book1.setstaftakenuntil(sc.nextLine());
        }
    } else if (is_book2 && book2.grabstatus().equals("borrowed")) {
        if (Ahmad_logged_in) {
            book2.setstudtakenuntil(sc.nextLine());
        } else {
            book2.setstaftakenuntil(sc.nextLine());
        }
    } else if (is_book3 && book3.grabstatus().equals("borrowed")) {
        if (Ahmad_logged_in) {
            book3.setstudtakenuntil(sc.nextLine());
        } else {
            book3.setstaftakenuntil(sc.nextLine());
        }
    } else {
        System.out.println("Error");
```

```
    }
    break;
}

case "7" -> {
    if (is_book1 && book1.grabstatus().equals("borrowed")) {
        book1.setstatus("available");
        book1.setoverduedate("none");
        if (Ahmad_logged_in) {
            Ahmad.dropborrowing(book1.gettitle());
            Ahmad.setstudentlimit(0);
        } else {
            Damian.dropborrowing(book1.gettitle());
            Damian.setstafflimit(0);
        }
    } else if (is_book2 && book2.grabstatus().equals("borrowed")) {
        book2.setstatus("available");
        book2.setoverduedate("none");
        if (Ahmad_logged_in) {
            Ahmad.dropborrowing(book2.gettitle());
            Ahmad.setstudentlimit(0);
        } else {
            Damian.dropborrowing(book2.gettitle());
            Damian.setstafflimit(0);
```

```
            }
        } else if (is_book3 && book3.grabstatus().equals("borrowed")) {
            book3.setstatus("available");
            book3.setoverduedate("none");
            if (Ahmad_logged_in) {
                Ahmad.dropborrowing(book3.gettitle());
                Ahmad.setstudentlimit(0);
            } else {
                Damian.dropborrowing(book3.gettitle());
                Damian.setstafflimit(0);
            }
        } else {
            System.out.println("Error");
        }
        break;
    }

    case "8" -> {
        if (is_book1 && book1.grabstatus().equals("borrowed")) {
            book1.setreservation(sc.nextLine());
            book1.setstatus("reserved");
        } else if (is_book2 && book2.grabstatus().equals("borrowed")) {
            book2.setreservation(sc.nextLine());
            book2.setstatus("reserved");
```

```
            } else if (is_book3 && book3.grabstatus().equals("borrowed")) {
                book3.setreservation(sc.nextLine());
                book3.setstatus("reserved");
            } else {
                System.out.println("Error");
            }
            break;
        }

        case "9" -> {
            if (is_book1) {
                System.out.println(book1.get_days_till_overdue() + " days till
overdue.");
            } else if (is_book2) {
                System.out.println(book2.get_days_till_overdue() + " days till
overdue.");
                book2.getstatus();
            } else if (is_book3) {
                System.out.println(book3.get_days_till_overdue() + " days till
overdue.");
                book3.getstatus();
            }
            break;
        }
```

```
                case "10" -> {
                    if (is_Ahmad) {
                        System.out.println("Still can borrow: " + Ahmad.getstudlimit());
                    } else {
                        System.out.println("Still can borrow: " + Damian.getstafflimit());
                    }
                    break;
                }
                case "11" -> {
                    if (is_Ahmad) {
                        System.out.println("Currently borrowing: " + Ahmad.getborrowing());
                        System.out.println("Borrow history: " + Ahmad.grabhistory());
                        System.out.println("Still can borrow: " + Ahmad.getstudlimit());
                        System.out.println("Currently charged for: RM" +
Ahmad.grabcharges());
                    } else {
                        System.out.println("Currently borrowing: " + Damian.getborrowing());
                        System.out.println("Borrow history: " + Damian.grabhistory());
                        System.out.println("Still can borrow: " + Damian.getstafflimit());
                        System.out.println("Currently charged for: RM" +
Damian.grabcharges());
                    }
                    break;
                }
                case "12" -> {
```

```
                    System.out.println("Thank you for using our system.");
                    System.exit(0);
                    break;
                }
            }


        }
    } else {
        System.out.println("Error");
    }
    System.out.println("\n\nDo you wish to continue using this account?");
    answer = sc.nextLine();
  } while (answer.equals("yes"));
 } while (answer.equals("no"));
 }
}
```

Library.java

```
package library_management_system;


import java.text.ParseException;
import java.time.format.DateTimeFormatter;
import java.time.LocalDate;
import java.time.temporal.ChronoUnit;


public class Library {


    public String title;
    public String author;
    public String category;
    public String synopsis = "No one has write a synopsis yet.";
    public String status = "available";
    public String overduedate = "-";
    public String takenby;
    public String takenuntil;
    public int days_till_overdue;
    public String reservedby;
    public int numberborrowed;


    public Library(String booktitle, String bookauthor, String bookcategory) {
        title = booktitle;
```

```
    author = bookauthor;
    category = bookcategory;
}

public String gettitle() {
    return this.title;
}

public String getauthor() {
    return this.author;
}

public String getcategory() {
    return this.category;
}

public void getstatus() {
    System.out.println("Status: " + this.status);
}

public String grabstatus() {
    return this.status;
}
```

```
public void getoverduedate() {
    System.out.println("Overdue by: " + this.overduedate);
}

public void getsynopsis() {
    System.out.println("Synopsis: " + this.synopsis);
}

public void getdetails() {
    System.out.println("Title: " + gettitle());
    System.out.println("Author: " + getauthor());
    System.out.println("Category: " + getcategory());
    System.out.println("Status: " + status);
}

public void settitle(String title) {
    this.title = title;
}

public void setauthor(String author) {
    this.author = author;
}

public void setcategory(String category) {
```

```
    this.category = category;
}

public void setstatus(String status) {
    this.status = status;
}

public void setoverduedate(String date) {
    this.overduedate = date;
}

public void setsynopsis(String synopsis) {
    this.synopsis = synopsis;
}

public void settakenby(String takenby) {
    this.takenby = takenby;
}

public String gettakenby() {
    return takenby;
}

public void setstudtakenuntil(String date) throws ParseException {
```

```
    if (date.equals("-")) {
        days_till_overdue = 0;
    } else {
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("d/MM/yyyy");
        LocalDate a = LocalDate.now();
        String b = date;
        LocalDate c = LocalDate.parse(b, formatter);
        long result = ChronoUnit.DAYS.between(a, c);
        days_till_overdue = (int) result;
        if (days_till_overdue < 30) {
            days_till_overdue = (int) result;
        } else {
            System.out.println("Exceeded 30 days. \n System will shut down.");
            System.exit(0);
        }
    }
}

public void setstaftakenuntil(String date) throws ParseException {
    if (date.equals("-")) {
        days_till_overdue = 0;
    } else {
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("d/MM/yyyy");
        LocalDate a = LocalDate.now();
```

```
        String b = date;
        LocalDate c = LocalDate.parse(b, formatter);
        long result = ChronoUnit.DAYS.between(a, c);
        days_till_overdue = (int) result;
        if (days_till_overdue < 90) {
            days_till_overdue = (int) result;
        } else {
            System.out.println("Exceeded 90 days. \n System will shut down.");
            System.exit(0);
        }
    }
}

public int get_days_till_overdue() {
    return days_till_overdue;
}

public void setreservation(String bywho) {
    reservedby = bywho;
}

public void setnewnumber() {
    this.numberborrowed++;
}
```

```
    public void getborrownumber() {
        System.out.println("This book has been borrowed for " + this.numberborrowed + " times.");
    }


}
```

User.java

```
package library_management_system;


import java.util.ArrayList;


public class User {


    String username;
    String usertype;
    String userpassword;
    int charged_amount;
    ArrayList<String> borrowed = new ArrayList<>();
    ArrayList<String> borrowing = new ArrayList<>();
    int studentbookcountlimit = 5;
    int staffbookcountlimit = 10;


    public User(String name, String type, String password) {
        setusername(name);
        setusertype(type);
        setuserpassword(password);
    }


    public void setusername(String name) {
        username = name;
```

```
}

public void setusertype(String type) {
    usertype = type;
}

public void setuserpassword(String password) {
    userpassword = password;
}

public String getusername() {
    return this.username;
}

public String getusertype() {
    return this.usertype;
}

public String getuserpassword() {
    return this.userpassword;
}

public void getuserdetails() {
    System.out.println("Username: " + getusername());
```

```
        System.out.println("User type: " + getusertype());
        System.out.println("Password: " + getuserpassword());
        System.out.println("Book(s) borrowed: " + borrowed);
        System.out.println("Total charges: " + charged_amount);
    }

    public void addcharges(int amount) {
        this.charged_amount += amount;
    }

    public void getcharges() {
        System.out.println("This person has been charged RM" + charged_amount + " till
date.");
    }

    public int grabcharges() {
        return charged_amount;
    }

    public void addbook(String book) {
        borrowed.add(book);
    }

    public void gethistory() {
        System.out.println("Book(s) borrowed: " + borrowed);
```

```
}

public ArrayList grabhistory() {
    return borrowed;
}

public void setborrowing(String bookname) {
    borrowing.add(bookname);
}

public void dropborrowing(String bookname) {
    borrowing.remove(bookname);
}

public ArrayList getborrowing() {
    return borrowing;
}

public void getcurrentbookborrowed() {
    System.out.println("Borrowing: " + borrowing);
}

public int getstudlimit() {
    return studentbookcountlimit;
```

```
    }

    public int getstafflimit() {
        return staffbookcountlimit;
    }

    public void setstudentlimit(int a) {
        if (a == 1) {
            studentbookcountlimit--;
        } else {
            studentbookcountlimit++;
        }
    }

    public void setstafflimit(int a) {
        if (a == 1) {
            staffbookcountlimit--;
        } else {
            staffbookcountlimit++;
        }
    }

}
```