

```

public class Books {
    int bookID,bookQuantity;
    static int serial=1001;
    //To give id to the books
    String bookName,authorName,publisher;
    Books(String bookName,String authorName,String publisher) {
        //to add initial books
        bookID=serial++;
        this.bookName=bookName;
        this.authorName=authorName;
        this.bookQuantity= 5;
        this.publisher=publisher;
    }
    Books() {
        //do nothing constructor
    }
    void addBook(int i) {
        //To add book entered by the librarian
        bookID=1001+i;
        System.out.print("Enter Book Name : ");
        bookName=Library.scan.nextLine();
        System.out.print("Enter Author Name : ");
        authorName=Library.scan.nextLine();
        System.out.print("Enter Publisher : ");
        publisher=Library.scan.nextLine();
        System.out.print("Enter Quantity : ");
        bookQuantity=Library.scan.nextInt();
    }
    void display() {
        //to display all the books
        System.out.print(bookID+" ");
        System.out.print(bookName);
        spaceBookName();
        System.out.print(authorName);
        spaceAuthorName();
        System.out.print(publisher);
        spacePublisher();
        System.out.print(" "+bookQuantity+"\n");
    }
    void header() {
        //Label used when we want to display all books
        System.out.println(" ID      Book Name      "+"      Author Name      "+"  

Publisher      "+"      Quantity ");
    }
    void displayOneBook() {
        //to display one single book
        System.out.println("ID      : "+bookID+"\nBook Name : "+bookName+"\nAuthor Name : "+  

authorName+"\nPublisher : "+publisher);
    }
    //these three methods are adding extra space between two values so that books are displayed in a  

better way
    void spaceBookName() {

```

```
int space=39-bookName.length();
if(space<0){
    System.out.print("    ");
} else {
    int i = 0;
    while (i < space) {
        System.out.print(" ");
        i++;
    }
}
void spaceAuthorName() {
    int space=25-authorName.length();
    if(space<0){
        System.out.print("    ");
    } else {
        int i = 0;
        while (i < space) {
            System.out.print(" ");
            i++;
        }
    }
}
void spacePublisher() {
    int space=25-publisher.length();
    if(space<0){
        System.out.print("    ");
    } else {
        int i = 0;
        while (i < space) {
            System.out.print(" ");
            i++;
        }
    }
}
```

```

import java.util.*;
public class Library {
    static Scanner scan=new Scanner(System.in);
    static Random random = new Random();
    static Librarian[] librarians=new Librarian[5]; //creates 5 librarians
    static Student[] students=new Student[50]; //creates 50 students
    static Books[] book=new Books[100]; //creates 100 books
    /*
     * A constructor to set
     * --> Default librarian
     *      --> User ID : Librarian@21
     *      --> Password : Librarian@21
     * --> Store 10 books data
     * --> Register 5 students data
     */
    Library() {
        librarians[0]=new Librarian("Librarian","Librarian@21","21");
        for(int a=1;a< librarians.length;a++) {
            librarians[a]=new Librarian();
        }
        // Already registered students
        students[0]=new Student("Akash Rai","22002171310109");
        students[1]=new Student("Mansi Malaviya","22002171310054");
        students[2]=new Student("Aditya Singh","22002171210065");
        students[3]=new Student("Patel Hensy","22002171310082");
        students[4]=new Student("Bhatt Shreeom","22002171310008");
        for(int i=5;i< students.length;i++) {
            students[i]=new Student();
        }
        // Some books which are already added to library
        book[0]=new Books("Java The Complete Reference","Herbert Schildt","Tata McGrawHill");
        book[1]=new Books("Core Java An Integrated Approach","Dr R Nageswara Rao","Dream tech
Press");
        book[2]=new Books("Core Java Volume I-Fundamentals","Cay S. Horstmann","Pearson India");
        book[3]=new Books("The Java Programming Language","James Gosling","Addison Wesley");
        book[4]=new Books("Higher Engineering Mathematics","Dr B S Grewal","Khanna Publisher");
        book[5]=new Books("Engineering Physics","G Vijayakumari","Vikas Publishing House");
        book[6]=new Books("Software Engineering","Roger S. Pressman","Tata McGrawHill");
        book[7]=new Books("Beginning Arduino","Michael McRoberts","Apress");
        book[8]=new Books("Environmental Science","Dr. B.R. Shah","Mahajan Publishing House");
        book[9]=new Books("Physics-2","Nimish Das","Books India");
        for(int i=10;i< book.length;i++) {
            book[i]=new Books();
        }
    }
    static int librarianAccount_count =1; //counts how many librarians account have been created
    int choice; //stores choices entered by the user
    int loginAttempt_count; //counts how many login attempt had been made by the user
    boolean access; //stores true--> login id and password are correct
    void loginMenu() {
        //login menu
        System.out.println("*****");
    }
}

```

```

System.out.println("Press 0 to Exit Application.");
System.out.println("Press 1 Sign Up.");
System.out.println("Press 2 Login.");
System.out.println("*****");
choice=scan.nextInt();
scan.nextLine();
switch (choice) {
    case 0 : return;
    case 1 : signUp();
        loginMenu();
        break;
    case 2 : login();
        break;
    default: System.out.println("Invalid Input\nPress between 0 to 2");
        loginMenu();
}
}

//signUp method
void signUp() {
    if(librarianAccount_count == librarians.length) {
        System.out.println("Maximum Accounts Created");
    } else {
        librarians[librarianAccount_count].signUp();
        //calls the signUp method in the Librarians class to register a librarian
        librarianAccount_count++;
    }
}

void login() {
    // login method
    System.out.print("Enter User ID : ");
    String enteredUserId = scan.nextLine();
    System.out.print("Enter Password : ");
    String enteredPassword = scan.nextLine();
    captcha();
    if (loginAttempt_count == 2) {
        System.out.println("Account locked for 1 hour");
        access = false;
    } else {
        for (int i = 0; i < librarians.length; i++) {
            if (librarians[i].librarianId == null) {
                System.out.println("Incorrect User Id or Password");
                loginAttempt_count++;
                scan.nextLine();
                login();
                break;
            } else {
                librarians[i].login(enteredUserId, enteredPassword);
                if (librarians[i].access) {
                    access = true;
                    break;
                } else if (i == (librarians.length-1)) {
                    System.out.println("Incorrect User Id or Password");
                }
            }
        }
    }
}

```

```

        loginAttempt_count++;
        login();
    }
}
}
}
}

void captcha() {
    // To generate Captcha
    int a=random.nextInt(100); //creates a random variable between 0 and 99
    int b=random.nextInt(10); //creates a random variable between 0 and 9
    int operator=random.nextInt(3);
    char operator_sign;
    int answer;
    if (operator == 1) {
        operator_sign = '-';
        answer = a - b;
    } else if(operator == 2) {
        operator_sign = '+';
        answer = a + b;
    } else {
        operator_sign = '*';
        answer = a * b;
    }
    System.out.print("CAPTCHA --> "+a+operator_sign+b+" = ");
    int ans=scan.nextInt();
    if(answer!=ans) {
        System.out.println("Entered Captcha answer is incorrect\n");
        captcha();
    }
}

void control() {
    // To execute all methods as per user input
    mainMenu();
    if(mainMenu_choice ==0) {
        loginMenu();
        if(choice==0||!access){
            if(choice==0) {
                System.out.println("***** Thanks for using our system *****");
            }
            return;
        }
    } else if(mainMenu_choice ==11) {
        showLibrarians();
    }
    control();
}

void showLibrarians() {
    // To show all signed up librarians
    librarians[0].header();
    for (Librarian librarian : librarians) {
        if (librarian.librarianId == null) {

```

```

        continue;
    }
    librarian.display();
}
}

int mainMenu_choice, bookId;
void mainMenu() {
    // All functionalities
    System.out.println("\n*****");
    System.out.println("Press 0 to return to the login menu");
    System.out.println("Press 1 to Add new Book");
    System.out.println("Press 2 to Upgrade Quantity of a Book");
    System.out.println("Press 3 to Search a Book");
    System.out.println("Press 4 to Show All Books");
    System.out.println("Press 5 to Delete a Book");
    System.out.println("Press 6 to Register Student");
    System.out.println("Press 7 to Show All Registered Students");
    System.out.println("Press 8 to issue Book");
    System.out.println("Press 9 to return Book");
    System.out.println("Press 10 to search a student");
    System.out.println("Press 11 to Show All Registered Librarians");
    System.out.println("*****");
    mainMenu_choice = scan.nextInt();
    switch (mainMenu_choice) {
        case 0:
        case 11:
            return;
        case 1: no_Books();
            break;
        case 2: upgradeQuantity();
            break;
        case 3: searchBooks();
            break;
        case 4: showBooks();
            break;
        case 5 : deleteBook();
            break;
        case 6: registerStudent();
            break;
        case 7: showStudents();
            break;
        case 8: issueBook(searchStudent());
            break;
        case 9: returnBook(searchStudent());
            break;
        case 10: searchStudent();
            break;
        default: System.out.println("Invalid Input\nPress between 0 to 11");
    }
    mainMenu();
}
void no_Books() {
}

```

```

// To check whether there is space for new book or not
int freeSpace_count=0;
int nBook;
for (Books books : book) {
    if (books.bookName == null) {
        freeSpace_count++;
    }
}
if(freeSpace_count==0) {
    System.out.println("Shelf is full");
    return;
}
for(;;) {
    System.out.print("Enter no of books to be added : ");
    nBook= scan.nextInt();
    if(nBook<=freeSpace_count) {
        break;
    }
    System.out.println("Shelf can now only store : "+freeSpace_count+" books");
}
for(int i=0;i<nBook;i++) {
    System.out.println("Enter Book "+(i+1));
    addBooks();
    System.out.println();
}
System.out.println("Books are added successfully");
}

void addBooks() {
    // To add book
    scan.nextLine();
    for (int i=0;i<book.length;i++) {
        if (book[i].bookName == null) {
            book[i].addBook(i);
            break;
        }
    }
}

void showBooks() {
    // To display all books
    book[0].header();
    for (Books books : book) {
        if (books.bookName == null) {
            continue;
        }
        books.display();
    }
}

void upgradeQuantity() {
    // To upgrade quantity of available books
    System.out.println("To Upgrade Quantity");
    bookId=searchID();
    if(bookId==401) {

```

```

        System.out.println("No such book found in the library");
        return;
    }
    System.out.print("Quantity you want to Add : ");
    int quantity=scan.nextInt();
    book[bookId].bookQuantity+=quantity;
    System.out.println("Quantity has been upgraded successfully");
}

void searchBooks() {
    // To search a book from all added books in 4 ways
    System.out.println("*****");
    System.out.println("Press 1 to Search by Book ID");
    System.out.println("Press 2 to Search by Book Name");
    System.out.println("Press 3 to Search by Author Name");
    System.out.println("Press 4 to Search by publisher Name");
    System.out.println("*****");
    int search_choice= scan.nextInt();
    switch (search_choice) {
        case 1 -> bookId = searchID();
        case 2 -> searchName();
        case 3 -> searchAuthor();
        case 4 -> searchPublisher();
        default -> {
            System.out.println("Invalid Input\nPress between 1 to 4");
            searchBooks();
        }
    }
    if(search_choice==1) {
        if (bookId == 401) {
            System.out.println("No such book found in the library");
        }
        else {
            book[bookId].displayOneBook();
        }
    }
}
int searchID() {
    // Search a books by ID
    System.out.print("Enter Book ID : ");
    int enteredBookID=scan.nextInt();
    int i=0;
    for(;i<book.length;i++) {
        if(book[i].bookID==0){
            return 401;//error code
        }
        else if(book[i].bookID==enteredBookID){
            break;
        }
    }
    return i;
}
void searchName() {
}

```

```

//To search a book by name
System.out.print("Enter Book Name : ");
scan.nextLine();
String enteredBookName=scan.nextLine();
int display_count=0;
for (Books books : book) {
    if (books.bookName == null) {
        if (display_count == 0) {
            System.out.println("No such book found in the library");
        }
        break;
    } else if (books.bookName.equalsIgnoreCase(enteredBookName)) {
        books.displayOneBook();
        System.out.println();
        display_count++;
    }
}
void searchAuthor() {
    // To search a book by Author name
    System.out.print("Enter Author Name : ");
    scan.nextLine();
    String enteredAuthorName=scan.nextLine();
    int display_count=0;
    for (Books books : book) {
        if (books.authorName == null) {
            if (display_count == 0) {
                System.out.println("No such book found in the library");
            }
            break;
        } else if (books.authorName.equalsIgnoreCase(enteredAuthorName)) {
            books.displayOneBook();
            System.out.println();
            display_count++;
        }
    }
}
void searchPublisher() {
    // To search a book by publisher name
    System.out.print("Enter Publisher : ");
    scan.nextLine();
    String enteredPublisher=scan.nextLine();
    int display_count=0;
    for (Books books : book) {
        if (books.publisher == null) {
            if (display_count == 0) {
                System.out.println("No such book found in the library");
            }
            break;
        } else if (books.publisher.equalsIgnoreCase(enteredPublisher)) {
            books.displayOneBook();
            System.out.println();
        }
    }
}

```

```

        display_count++;
    }
}
}

void deleteBook() {
    bookId = searchID();
    if (bookId == 401) {
        System.out.println("No such book found in the library");
    }
    else {
        book[bookId].bookName=null;
        book[bookId].authorName=null;
        book[bookId].publisher=null;
        book[bookId].bookQuantity=0;
        book[bookId].bookID=0;
        System.out.println("Book has been successfully deleted");
    }
}
void registerStudent(){
    // To register student
    System.out.print("Student enrollment number : ");
    scan.nextLine();
    String enrollmentNumber=scan.nextLine();
    int check=0;
    for (Student student : students) {
        if (student.studentName == null) {
            break;
        } else if (student.enrollmentNumber.equals(enrollmentNumber)) {
            System.out.println("Student already registered ");
            check = 1;
            break;
        }
    }
    if(check==0) {
        for (Student student : students) {
            if (student.studentName == null) {
                student.registration(enrollmentNumber);
                System.out.println("Student registered successfully");
                break;
            }
        }
    }
}
void showStudents() {
    // To display all Students
    students[0].header();
    for (Student student : students) {
        if (student.studentName == null) {
            break;
        }
        student.displayStudent();
    }
}

```

```

}

int searchStudent() {
    // To search student records by id
    System.out.print("Enter Student Id : ");
    scan.nextLine();
    String id=scan.nextLine();
    for (int i=0;i< students.length;i++) {
        if(students[i].libID==null) {
            System.out.println("ID doesn't exist");
            break;
        }
        else {
            if (students[i].libID.equals(id)) {
                System.out.println("ID verified");
                students[i].displayOneStudent();
                return i;
            }
            else if (i == 49) {
                System.out.println("ID doesn't exist");
            }
        }
    }
    return 401;
}

void issueBook(int student_id) {
    // To issue book
    if(student_id!=401) {
        int issuedBook_count=students[student_id].count();
        if(issuedBook_count<3) {
            System.out.println();
            bookId = searchID();
            while (bookId == 401) {
                System.out.println("Please enter correct book id");
                bookId = searchID();
            }
            book[bookId].displayOneBook();
            if (book[bookId].bookQuantity > 0) {
                scan.nextLine();
                System.out.print("Enter issue date(DD MM YYYY) : ");
                String issueDate = scan.nextLine();
                while(checkDate(issueDate)){
                    System.out.println("Not a valid date re-enter date");
                    System.out.print("Enter issue date(DD MM YYYY) : ");
                    issueDate = scan.nextLine();
                }
                students[student_id].issueBook(book[bookId].bookID, issueDate);
                book[bookId].bookQuantity--;
                System.out.println("Book issued successfully");
            } else {
                System.out.println("Book not available");
            }
        } else {

```

```

        System.out.println("Maximum no of book issued");
    }
}
}

void returnBook(int student_id) {
    // Return book
    if(student_id!=401) {
        System.out.println();
        bookId=searchID();
        while (bookId == 401) {
            System.out.println("Please enter correct book id");
            bookId = searchID();
        }
        if(students[student_id].checkBook(book[bookId].bookID)) {
            System.out.println();
            scan.nextLine();
            System.out.print("Enter return date(DD MM YYYY) : ");
            String returnDate=scan.nextLine();
            while(checkDate(returnDate)){
                System.out.println("Not a valid date re-enter date");
                System.out.print("Enter return date(DD MM YYYY) : ");
                returnDate=scan.nextLine();
            }
            while(!students[student_id].checkReturnDate(book[bookId].bookID,returnDate)){
                System.out.println("return date cannot come before issue date\nre-enter date");
                System.out.print("Enter return date(DD MM YYYY) : ");
                returnDate=scan.nextLine();
                while(checkDate(returnDate)){
                    System.out.println("Not a valid date re-enter date");
                    System.out.print("Enter return date(DD MM YYYY) : ");
                    returnDate=scan.nextLine();
                }
            }
            int days=students[student_id].returnBook(book[bookId].bookID,returnDate);
            if(days>15) {
                System.out.println(students[student_id].studentName+" is late for returning book by "+(days-15)+" days\nFine : " +((days-15)*2));
            }
            System.out.println("Book returned successfully");
            book[bookId].bookQuantity++;
        } else {
            System.out.println("Student has issued no such book");
        }
    }
}

boolean checkDate(String a) {
    int count=0;
    for(int i=0;i<a.length();i++) {
        if(a.charAt(i)==' ') {
            count++;
        }
    }
}

```

```
if(count!=2){
    return true;
}
String [] date=a.split(" ");
int day=Integer.parseInt(date[0]); //day
int month=Integer.parseInt(date[1]); //month
int year=Integer.parseInt(date[2]); //year
return (year <= 1000 || month <= 0 || month > 12 || day <= 0 || day > 31 || a.length() != 10);
}
public static void main(String[]args) {
    Library library=new Library();
    System.out.println("***** Welcome to the Library *****");
    library.loginMenu();
    if(library.choice==0||!library.access){
        if(library.choice==0) {
            System.out.println("***** Thanks for using our system *****");
        }
        return;
    }
    library.control();
}
}
```

```

public class Student {
    String studentName,enrollmentNumber;
    String libID; // Student's library id
    int[] issuedBookID={0,0,0}; // Array of issued books
    String [] issueDate={null,null,null}; // Array of issue dates
    String [] returnDate={null,null,null}; // Array of return date
    static int serial=1001;
    Student(String studentName,String enrollmentNumber) {
        libID="S"+serial++;
        this.studentName=studentName;
        this.enrollmentNumber=enrollmentNumber;
    }
    Student() {
        // Do nothing constructor
    }
    void registration(String enrollmentNumber) {
        // To give student id and add student name
        this.enrollmentNumber=enrollmentNumber;
        libID="S"+serial;
        serial++;
        System.out.print("Student name : ");
        studentName=Library.scan.nextLine();
    }
    void displayStudent() {
        // To display all Students
        System.out.print(libID+" ");
        System.out.print(enrollmentNumber);
        spaceEnroll();
        System.out.print(studentName+"\n");
    }
    void header() {
        System.out.println(" ID "+"Enrollment Number "+" Student name");
    }
    void spaceEnroll() {
        // To add some extra space for better looks
        int space=22-enrollmentNumber.length();
        if(space<0){
            System.out.print("   ");
        } else {
            int i = 0;
            while (i < space) {
                System.out.print(" ");
                i++;
            }
        }
    }
    void displayOneStudent() {
        // To display details of searched student id
        System.out.println("Id : "+libID+"\nEnrollment Number : "+enrollmentNumber+"\nName : "+
studentName);
        System.out.println();
        System.out.println("Issued Book");
    }
}

```

```

int count=0;
for(int i=0;i<3;i++) {
    if(issuedBookID[i]!=0) {
        for(int j=0;j<100;j++) {
            if(Library.book[j].bookID==issuedBookID[i]) {
                System.out.println("Book ID : "+ Library.book[j].bookID+ " Book Name :" + Library.
book[j].bookName);
                break;
            }
        }
        count++;
    }
}
if(count==0) {
    System.out.println("None");
}
}

int count(){
    // To count number of issued books
    int countIssuedBook=0;
    for(int i=0;i<3;i++) {
        if(issuedBookID[i]!=0) {
            countIssuedBook++;
        }
    }
    return (countIssuedBook);
}

void issueBook(int id,String date) {
    // To add book id to issued book and take issue date
    for(int i=0;i<3;i++) {
        if(issuedBookID[i]==0) {
            issuedBookID[i]=id;
            issueDate[i]=date;
            break;
        }
    }
}

boolean checkBook(int id) {
    // To check book id in issued books
    for(int i=0;i<3;i++) {
        if(issuedBookID[i]==id) {
            return true;
        }
    }
    return false;
}

int returnBook(int id,String date) {
    // To remove book from issued book and take return date
    for(int i=0;i<3;i++) {
        if(issuedBookID[i]==id) {
            issuedBookID[i]=0;
            returnDate[i]=date;
        }
    }
}

```

```

        int days = calculateNoOfDays(i);
        returnDate[i] = null;
        return (days);
    }
}
return 0;
}
boolean checkReturnDate(int id, String returnDate) {
    int i;
    for(i=0;i<3;i++) {
        if(issuedBookID[i]==id) {
            break;
        }
    }
    String [] issue=issueDate[i].split(" ");
    int issueDay=Integer.parseInt(issue[0]);
    int issueMonth=Integer.parseInt(issue[1]);
    int issueYear=Integer.parseInt(issue[2]);
    String [] returnD=returnDate.split(" ");
    int returnDay=Integer.parseInt(returnD[0]);
    int returnMonth=Integer.parseInt(returnD[1]);
    int returnYear=Integer.parseInt(returnD[2]);
    if(returnYear>issueYear) {
        return true;
    } else if (returnYear==issueYear) {
        if(returnMonth>issueMonth) {
            return true;
        } else if (returnMonth==issueMonth) {
            return returnDay >= issueDay;
        } else {
            return false;
        }
    } else {
        return false;
    }
}
// Two methods to calculate number of days between issue date and return date
int calculateNoOfDays(int i) {
    int days;
    String [] issue=issueDate[i].split(" ");
    int issueDay=Integer.parseInt(issue[0]);
    int issueMonth=Integer.parseInt(issue[1]);
    int issueYear=Integer.parseInt(issue[2]);
    String [] returnD=returnDate[i].split(" ");
    int returnDay=Integer.parseInt(returnD[0]);
    int returnMonth=Integer.parseInt(returnD[1]);
    int returnYear=Integer.parseInt(returnD[2]);
    boolean issueLeapYear= (issueYear%4==0 && issueYear%100!=0)||((issueYear%400==0));
    boolean returnLeapYear= (returnYear%4==0 && returnYear%100!=0)||((returnYear%400==0));
    if (issueMonth == returnMonth && issueYear==returnYear) {
        days = returnDay - issueDay +1;
    } else {

```

```

int extraMonthDays=0;
int issueMonthDays=calculateDaysOfMonths(issueMonth,issueLeapYear)-issueDay+1;
if(issueYear==returnYear) {
    if (issueMonth+1 != returnMonth) {
        for (int month = issueMonth+1; month < returnMonth; month++) {
            extraMonthDays+=calculateDaysOfMonths(month,issueLeapYear);
        }
    }
} else {
    int years=returnYear-issueYear-1;
    for(int year=1;year<=years;year++) {
        if(((issueYear+year)%4==0 && (issueYear+year)%100!=0)||((issueYear+year)%400==0)) {
            extraMonthDays+=366;
        } else {
            extraMonthDays+=365;
        }
    }
    for (int month = issueMonth+1; month <=12; month++) {
        extraMonthDays+=calculateDaysOfMonths(month,issueLeapYear);
    }
    for (int month = 1; month <returnMonth; month++) {
        extraMonthDays+=calculateDaysOfMonths(month,returnLeapYear);
    }
}
days=returnDay+issueMonthDays+extraMonthDays;
}
return days;
}

int calculateDaysOfMonths(int month,boolean leapYear) {
    int day=0;
    switch (month) {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            day += 31;
            break;
        case 2:
            if (leapYear) {
                day += 29;
            }
            else {
                day+= 28;
            }
            break;
        case 4:
        case 6:
        case 9:
        case 11:
    }
}

```

```
        day += 30;
        break;
    }
    return day;
}
}
```

```

public class Librarian {
    String password,fullName,librarianId,age;
    boolean access;
    Librarian(String fullName,String password,String age) {
        this.fullName=fullName;
        this.password=password;
        this.age=age;
        createLibrarianID();
    }
    Librarian(){
        //do nothing constructor
    }

    void signUp() {
        // To signup
        System.out.print("Enter Full Name : ");
        fullName=Library.scan.nextLine();
        System.out.print("Enter Age : ");
        age=Library.scan.nextLine();
        int temp_age=Integer.parseInt(age);
        if(temp_age<=20) {
            System.out.println("Your age must be above 20 to sign up");
            return;
        }
        enterPassword();
        createLibrarianID();
        System.out.println("You have successfully Signed Up \nYour login Id is : "+librarianId);
    }
    void enterPassword() {
        // To take password as input and verify it
        String pass;
        for(;;) {
            System.out.print("Enter Password : ");
            pass =Library.scan.nextLine();
            int countDigit=0,countSpecial=0,countLowercase=0,countUppercase=0;
            for(int i=0;i<pass.length();i++) {
                if(pass.charAt(i)>='0' && pass.charAt(i)<='9') {
                    countDigit++;
                } else if (pass.charAt(i)>='a' && pass.charAt(i)<='z') {
                    countLowercase++;
                } else if (pass.charAt(i)>='A' && pass.charAt(i)<='Z') {
                    countUppercase++;
                } else {
                    countSpecial++;
                }
            }
            if(pass.length()>=8 && countDigit>=1 && countLowercase>=1 && countUppercase>=1 &&
countSpecial >=1) {
                break;
            } else {
                System.out.println("Please enter a password having :-"

```

```

--> Minimum length 8
--> Includes at least one lowercase letter
--> Includes at least one Uppercase letter
--> Includes at least one number
--> Includes at least one special character

        re-enter password""");
    }
}
System.out.print("Confirm Password : ");
String confirmPass=Library.scan.nextLine();
if(pass.equals(confirmPass)){
    password=pass;
} else {
    System.out.println("Password and confirm password don't match\nre-enter password");
    enterPassword();
}
}

void createLibrarianID() {
    // Create a user id for librarian
    String[] name=fullName.split(" ");
    librarianId=name[0]+"@";
    librarianId=librarianId.concat(age);
}

void login(String enteredUserId,String enteredPassword) {
    // To sign in userid
    if(enteredUserId.equals(librarianId)&&enteredPassword.equals(password)) {
        System.out.println("You have successfully signed in");
        access=true;
    } else {
        access=false;
    }
}

void display() {
    // To display all signed up Librarians
    System.out.print(librarianId);
    idSpace();
    System.out.print(fullName);
    nameSpace();
    System.out.print(age+"\n");
}

// Adding some extra space for better looks
void idSpace(){
    int space=15-librarianId.length();
    if(space<0){
        System.out.print("    ");
    } else {
        int i = 0;
        while (i < space) {
            System.out.print(" ");
            i++;
    }
}

```

```
        }
    }
}

void nameSpace(){
    int space=30fullName.length();
    if(space<0){
        System.out.print("    ");
    } else {
        int i = 0;
        while (i < space) {
            System.out.print(" ");
            i++;
        }
    }
}

void header() {
    //Label used when we want to display all librarians
    System.out.println("Librarian ID  Librarian Name          Librarian Age");
}
}
```