Assignment -1

**1. Write a Java Program to create a Book class. Get all the values of attributes of a several books through an input screen (Use AWT or Swings, EventHandling for cursor movement & validations. The User Interface should be independent of the backend datastore i.e. file or DB). Save the data to a file, Books.dat. After saving all the records, close the file. Reopen the same data file, read it & display the list of all books, on screen (One should be able to scroll through all the list of records). The Book class should contain details related to bookId, bookName, authorNames, publication, dateOfPublication, priceOfBook, totalQuantityToOrder, totalCost.**

**2. Query a record on the basis of bookid or bookname or authorname or publication should be possible. The data be retrieved from the Books.dat file and displayed in GUI. Also update the records. Deletion of any book record should also be possible.**

**# Book.java :-**

package Assignment1;

public class Book

{

int bookId;

String bookName;

String authorNames;

String publication;

String dateOfPublication;

float priceOfBook;

int totalQuantityToOrder;

float totalCost;

Book(){}

Book(int bookId,String bookName,String authorNames,String publication,String dateOfPublication,float priceOfBook,int totalQuantityToOrder)

{

this.bookId = bookId;

this.bookName = bookName;

this.authorNames = authorNames;

this.publication = publication;

this.dateOfPublication = dateOfPublication;

this.priceOfBook = priceOfBook;

this.totalQuantityToOrder = totalQuantityToOrder;

this.totalCost = totalQuantityToOrder\*priceOfBook;

}

public int getBookId() {

return bookId;

}

public void setBookId(int bookId) {

this.bookId = bookId;

}

public String getBookName() {

return bookName;

}

public void setBookName(String bookName) {

this.bookName = bookName;

}

public String getAuthorNames() {

return authorNames;

}

public void setAuthorNames(String authorNames) {

this.authorNames = authorNames;

}

public String getPublication() {

return publication;

}

public void setPublication(String publication) {

this.publication = publication;

}

public String getDateOfPublication() {

return dateOfPublication;

}

public void setDateOfPublication(String dateOfPublication) {

this.dateOfPublication = dateOfPublication;

}

public float getPriceOfBook() {

return priceOfBook;

}

public void setPriceOfBook(float priceOfBook) {

this.priceOfBook = priceOfBook;

this.totalCost = priceOfBook\*getTotalQuantityToOrder();

}

public int getTotalQuantityToOrder() {

return totalQuantityToOrder;

}

public void setTotalQuantityToOrder(int totalQuantityToOrder) {

this.totalQuantityToOrder = totalQuantityToOrder;

this.totalCost = totalQuantityToOrder\*getPriceOfBook();

}

public float getTotalCost() {

return totalCost;

}

}

**# book\_list.dat :-**

1\*Advance Java\*Mammta Padole\*MSU\*31-12-2024\*5000.0\*5

2\*Life of CR\*Sahil Bhanderi\*Sahil Publication\*01-03-2030\*100.0\*200

3\*Data Structures\*Anjali Sharma\*R.K. Publishers\*15-06-2025\*2000.0\*10

4\*Algorithms\*Rajesh Patel\*Patel & Sons\*25-12-2027\*1500.0\*8

5\*Web Development\*Neha Mehta\*MSU\*10-11-2026\*0.0\*12

6\*Database Systems\*Karan Joshi\*Joshi Enterprises\*02-02-2031\*500.0\*20

7\*Computer Networks\*Suresh Chavda\*National Book House\*18-07-2028\*3500.0\*15

8\*Software Engineering\*Pooja Desai\*Desai Print\*30-04-2029\*1000.0\*5

9\*Operating Systems\*Ravi Shah\*Shah Publications\*21-08-2024\*700.0\*7

10\*Artificial Intelligence\*Priya Thakkar\*MSU\*03-05-2027\*0.0\*25

11\*Machine Learning\*Vishal Gupta\*Gupta Bookstore\*16-09-2025\*4500.0\*12

12\*Cloud Computing\*Meera Rathi\*Modern Prints\*29-01-2032\*900.0\*6

13\*Networking Fundamentals\*Arjun Jain\*Jain Press\*07-10-2026\*300.0\*18

14\*Computer Architecture\*Swati Kumar\*Kumar Publishing\*12-03-2028\*2200.0\*9

15\*Cyber Security\*Manoj Solanki\*Solanki Distributors\*08-12-2030\*5000.0\*4

16\*Digital Marketing\*Simran Kaur\*MSU\*23-02-2029\*0.0\*30

17\*Blockchain Technology\*Akash Verma\*Verma & Co.\*04-06-2024\*1800.0\*11

18\*Business Intelligence\*Divya Nair\*Nair Publications\*11-07-2025\*3500.0\*14

19\*Web Security\*Nikhil Soni\*Soni Print Hub\*28-09-2027\*750.0\*13

20\*Game Development\*Ritika Jain\*Jain Enterprises\*09-11-2029\*2500.0\*8

21\*Reignite\*Abdul Kalam\*Science Books\*12-02-2011\*500.0\*2

**# BookList.java :-**

package Assignment1;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.util.ArrayList;

import java.util.List;

import java.util.StringTokenizer;

public class BookList {

FileReader fileReader;

BufferedReader bufferedReader;

BookList()

{

try

{

fileReader = new FileReader("book\_list.dat");

bufferedReader = new BufferedReader(fileReader);

}

catch (FileNotFoundException e)

{

e.printStackTrace();

}

}

List<Book> readBookList()

{

List<Book> books = new ArrayList<>();

try

{

String line;

while ((line = bufferedReader.readLine()) != null)

{

StringTokenizer st = new StringTokenizer(line, "\*");

Book b = new Book(

Integer.parseInt(st.nextToken()),

st.nextToken(),

st.nextToken(),

st.nextToken(),

st.nextToken(),

Float.parseFloat(st.nextToken()),

Integer.parseInt(st.nextToken())

);

books.add(b);

}

bufferedReader.close();

fileReader.close();

}

catch (Exception exp)

{

exp.printStackTrace();

}

return books;

}

}

**# DisplayFrame.java :-**

package Assignment1;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.List;

public class DisplayFrame extends JFrame

{

List<Book> books;

JPanel mainPanel;

JScrollPane scrollPane;

JLabel headingLabel;

JButton addBookButton;

public DisplayFrame(List<Book> books)

{

this.books = books;

this.mainPanel = new JPanel();

this.headingLabel = new JLabel("Books Display", JLabel.CENTER);

this.addBookButton = new JButton("Add New Book");

scrollPane = new JScrollPane(mainPanel);

createFrame();

}

void createFrame()

{

createHeaderPanel();

createMainPanel();

createBookPanels();

setSize(800, 1000);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

add(scrollPane);

setVisible(true);

}

void createHeaderPanel()

{

JPanel headerPanel = new JPanel();

headerPanel.setLayout(new BorderLayout());

headerPanel.setBackground(new Color(245, 245, 250));

headingLabel.setFont(new Font("Arial", Font.BOLD, 28));

headingLabel.setForeground(new Color(0, 0, 0));

headerPanel.add(headingLabel, BorderLayout.CENTER);

addBookButton = new JButton("+");

addBookButton.setFont(new Font("Arial", Font.BOLD, 28));

addBookButton.setPreferredSize(new Dimension(50, 50));

addBookButton.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent e)

{

InsertionFrame insertionFrame = new InsertionFrame();

}

});

JPanel buttonPanel = new JPanel();

buttonPanel.setBackground(new Color(245, 245, 250));

buttonPanel.add(addBookButton);

headerPanel.add(buttonPanel, BorderLayout.EAST);

this.add(headerPanel, BorderLayout.NORTH);

}

void createMainPanel()

{

mainPanel.add(Box.createVerticalStrut(20));

mainPanel.setLayout(new BoxLayout(mainPanel, BoxLayout.Y\_AXIS));

mainPanel.setBackground(new Color(245, 245, 250));

mainPanel.setBorder(BorderFactory.createEmptyBorder(15, 15, 15, 15));

}

void createBookPanels()

{

for (Book book : books)

{

BookPanel bookPanel = new BookPanel(book);

mainPanel.add(bookPanel);

mainPanel.add(Box.createVerticalStrut(15));

}

}

}

**# InsertFrame.java :-**

package Assignment1;

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import javax.swing.\*;

public class InsertionFrame extends JFrame

{

JLabel headerLabel;

JPanel panel;

String labelText[] = {"Book ID:","Book Name:","Author Names:","Publication:","Date of Publication:","Price of Book:","Total Quantity to Order:"};

JLabel label[];

JTextField textField[];

JButton submitButton;

JPanel footerPanel;

public InsertionFrame()

{

this.headerLabel = new JLabel("Insert New Book Details", JLabel.CENTER);

this.panel = new JPanel();

this.label = new JLabel[labelText.length];

this.textField = new JTextField[labelText.length];

this.submitButton = new JButton("Insert Book");

this.footerPanel = new JPanel();

createFrame();

}

void createFrame()

{

this.setBackground(new Color(245, 239, 255));

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

this.setSize(700, 600);

this.setLayout(new BorderLayout(10, 10));

createComponents();

this.add(headerLabel, BorderLayout.NORTH);

addingComponentsToPanel();

this.add(panel, BorderLayout.CENTER);

this.add(footerPanel, BorderLayout.SOUTH);

this.setVisible(true);

}

void createComponents()

{

createHeader();

createPanel();

createLabel();

createSubmitButton();

createFooter();

}

void createHeader()

{

headerLabel.setFont(new Font("Arial", Font.BOLD, 24));

headerLabel.setBorder(BorderFactory.createEmptyBorder(10, 0, 10, 0));

}

void createPanel()

{

panel.setLayout(new GridLayout(7, 2, 10, 10));

panel.setBorder(BorderFactory.createEmptyBorder(20, 40, 20, 40));

}

void createLabel()

{

for(int i=0; i<label.length; i++)

{

label[i] = new JLabel(labelText[i]);

textField[i] = new JTextField();

}

}

void createSubmitButton()

{

submitButton.setForeground(Color.WHITE);

submitButton.setPreferredSize(new Dimension(300, 40));

submitButton.setBackground(new Color(146, 145, 194));

submitButton.setFont(new Font("Arial", Font.BOLD, 16));

submitButton.setFocusPainted(false);

submitButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

Book b = new Book(Integer.parseInt(textField[0].getText()), textField[1].getText(), textField[2].getText(), textField[3].getText(), textField[4].getText(), Float.parseFloat(textField[5].getText()), Integer.parseInt(textField[6].getText()));

try {

FileWriter fw = new FileWriter("book\_list.dat", true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write(b.getBookId() + "\*" + b.getBookName() + "\*" + b.getAuthorNames() + "\*" + b.getPublication() + "\*" + b.getDateOfPublication() + "\*" + b.getPriceOfBook() + "\*" + b.getTotalQuantityToOrder());

bw.newLine();

bw.close();

fw.close();

} catch (Exception exp) {

exp.printStackTrace();

}

showMessageDialog(b);

}});

}

void showMessageDialog(Book b)

{

JOptionPane.showMessageDialog(this, "Book Inserted:\n" +

"Book ID: " + b.getBookId() + "\n" +

"Book Name: " + b.getBookName() + "\n" +

"Author Names: " + b.getAuthorNames() + "\n" +

"Publication: " + b.getPublication() + "\n" +

"Date of Publication: " + b.getDateOfPublication() + "\n" +

"Price of Book: " + b.getPriceOfBook() + "\n" +

"Total Quantity to Order: " + b.getTotalQuantityToOrder() + "\n" +

"Total Cost: " + b.getTotalCost());

}

void createFooter()

{

footerPanel.setBorder(BorderFactory.createEmptyBorder(20, 0, 20, 0));

footerPanel.add(submitButton);

}

void addingComponentsToPanel()

{

for(int i=0; i<label.length; i++)

{

panel.add(label[i]);

panel.add(textField[i]);

}

}

}

**# UpdateFrame.java :-**

package Assignment1;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.util.List;

import javax.swing.BorderFactory;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class UpdateFrame extends JFrame

{

Book book;

List<Book> books;

BookList bookList;

File file;

JLabel headerLabel;

JPanel panel;

String labelText[] = {"Book ID:","Book Name:","Author Names:","Publication:","Date of Publication:","Price of Book:","Total Quantity to Order:"};

JLabel label[];

JTextField textField[];

JButton submitButton;

JPanel footerPanel;

public UpdateFrame(Book book)

{

this.book = book;

this.bookList = new BookList();

this.books = bookList.readBookList();

this.file = new File("book\_list.dat");

this.headerLabel = new JLabel("Book Details", JLabel.CENTER);

this.panel = new JPanel();

this.label = new JLabel[labelText.length];

this.textField = new JTextField[labelText.length];

this.submitButton = new JButton("Update Book");

this.footerPanel = new JPanel();

createFrame();

}

void createFrame()

{

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

this.setSize(700, 600);

this.setLayout(new BorderLayout(10, 10));

createComponents();

this.add(headerLabel, BorderLayout.NORTH);

addingComponentsToPanel();

this.add(panel, BorderLayout.CENTER);

this.add(footerPanel, BorderLayout.SOUTH);

this.setVisible(true);

}

void createComponents()

{

createHeader();

createPanel();

createLabel();

createSubmitButton();

createFooter();

}

void createHeader()

{

headerLabel.setFont(new Font("Arial", Font.BOLD, 24));

headerLabel.setBorder(BorderFactory.createEmptyBorder(10, 0, 10, 0));

}

void createPanel()

{

panel.setLayout(new GridLayout(7, 2, 10, 10));

panel.setBorder(BorderFactory.createEmptyBorder(20, 40, 20, 40));

}

void createLabel()

{

for(int i=0; i<label.length; i++)

{

label[i] = new JLabel(labelText[i]);

textField[i] = new JTextField();

}

createTextField();

}

void createTextField()

{

textField[0].setEditable(false);

textField[0].setText(""+book.getBookId());

textField[1].setText(""+book.getBookName());

textField[2].setText(""+book.getAuthorNames());

textField[3].setText(""+book.getPublication());

textField[4].setText(""+book.getDateOfPublication());

textField[5].setText(""+book.getPriceOfBook());

textField[6].setText(""+book.getTotalQuantityToOrder());

}

void createSubmitButton()

{

submitButton.setPreferredSize(new Dimension(300, 40));

submitButton.setBackground(new Color(146, 145, 194));

submitButton.setForeground(Color.WHITE);

submitButton.setFont(new Font("Arial", Font.BOLD, 16));

submitButton.setFocusPainted(false);

submitButton.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent e)

{

Book b = new Book(Integer.parseInt(textField[0].getText()), textField[1].getText(), textField[2].getText(), textField[3].getText(), textField[4].getText(), Float.parseFloat(textField[5].getText()), Integer.parseInt(textField[6].getText()));

updateBook(b);

showMessageDialog();

}

});

}

void showMessageDialog()

{

JOptionPane.showMessageDialog(this, "Book Updated Successfully...!");

}

void updateBook(Book b)

{

try

{

FileWriter fileWriter = new FileWriter(file);

BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);

for (Book bk : books)

{

if(bk.bookId == book.bookId) bk = b;

bufferedWriter.write(bk.getBookId() + "\*" +

bk.getBookName() + "\*" +

bk.getAuthorNames() + "\*" +

bk.getPublication() + "\*" +

bk.getDateOfPublication() + "\*" +

bk.getPriceOfBook() + "\*" +

bk.getTotalQuantityToOrder());

bufferedWriter.newLine();

}

bufferedWriter.close();

fileWriter.close();

}

catch (Exception e)

{

e.printStackTrace();

System.out.println("Error writing to file!");

}

}

void createFooter()

{

footerPanel.setBorder(BorderFactory.createEmptyBorder(20, 0, 20, 0));

footerPanel.add(submitButton);

}

void addingComponentsToPanel()

{

for(int i=0; i<label.length; i++)

{

panel.add(label[i]);

panel.add(textField[i]);

}

}

}

**# BookPanel.java :-**

package Assignment1;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\*;

class BookPanel extends JPanel

{

Book book;

JPanel detailsPanel;

JPanel buttonPanel;

JButton updateButton;

JButton deleteButton;

JLabel labels[];

BookPanel(Book book)

{

this.book = book;

setLayout(new BorderLayout(10, 5));

detailsPanel = new JPanel(new GridLayout(4, 2, 25, 15));

buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER,20,10));

labels = createLabels();

deleteButton = new JButton("Delete");

updateButton = new JButton("Update");

createBookPanel();

}

void createBookPanel()

{

setBackground(new Color(245, 239, 255));

setBorder(BorderFactory.createCompoundBorder(

BorderFactory.createLineBorder(new Color(173, 173, 209), 3),

BorderFactory.createEmptyBorder(20, 20, 20, 20)));

detailsPanel.setBackground(new Color(245, 239, 255));

buttonPanel.setBackground(new Color(245, 239, 255));

for (JLabel label : labels)

{

label.setFont(new Font("Arial", Font.BOLD, 14));

detailsPanel.add(label);

}

updateDeleteButton();

add(detailsPanel, BorderLayout.CENTER);

add(buttonPanel, BorderLayout.SOUTH);

}

void updateDeleteButton()

{

updateButton.setBackground(new Color(146, 145, 194));

deleteButton.setBackground(new Color(146, 145, 194));

updateButton.setFont(new Font("Arial", Font.BOLD, 14));

deleteButton.setFont(new Font("Arial", Font.BOLD, 14));

updateButton.setForeground(Color.WHITE);

deleteButton.setForeground(Color.WHITE);

updateButton.setPreferredSize(new Dimension(300, 40));

deleteButton.setPreferredSize(new Dimension(300, 40));

updateButton.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent e)

{

JOptionPane.showMessageDialog(null, "Update button clicked for Book ID: " + book.getBookId());

UpdateFrame updateFrame = new UpdateFrame(book);

}

});

deleteButton.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent e)

{

BookDeletion bookDeletion = new BookDeletion(book);

if(bookDeletion.showDeleteConfirmation()==JOptionPane.YES\_OPTION)

{

bookDeletion.deleteBook();

Container parent = getParent();

if (parent != null)

{

parent.remove(BookPanel.this);

parent.revalidate();

parent.repaint();

}

}

}

});

buttonPanel.add(updateButton);

buttonPanel.add(deleteButton);

}

JLabel[] createLabels()

{

return new JLabel[]

{

new JLabel("Book ID: " + book.getBookId()),

new JLabel("Book Name: " + book.getBookName()),

new JLabel("Author: " + book.getAuthorNames()),

new JLabel("Publication: " + book.getPublication()),

new JLabel("Date: " + book.getDateOfPublication()),

new JLabel("Price: " + book.getPriceOfBook()),

new JLabel("Quantity: " + book.getTotalQuantityToOrder()),

new JLabel("Total Cost: " + book.getTotalCost()),

};

}

public void refreshFrame()

{

this.revalidate();

this.repaint();

}

Book getBook()

{

return book;

}

}

**# BookDisplay.java :-**

package Assignment1;

import java.util.List;

public class BookDisplay

{

public static void main(String[] args)

{

BookList booklist = new BookList();

List<Book> books = booklist.readBookList();

DisplayFrame frame = new DisplayFrame(books);

}

}

**# BookInsertion.java :-**

package Assignment1;

public class BookInsertion

{

public static void main(String[] args)

{

InsertionFrame frame = new InsertionFrame();

}

}

**# BookDeletion.java :-**

package Assignment1;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.util.List;

import javax.swing.JOptionPane;

public class BookDeletion

{

BookList bookList;

File file;

List<Book> books;

Book book;

public BookDeletion(Book book)

{

this.book = book;

this.file = new File("book\_list.dat");

this.bookList = new BookList();

books = bookList.readBookList();

}

public int showDeleteConfirmation()

{

return JOptionPane.showConfirmDialog(

null,

"Are you sure you want to remove this book ?",

"Confirm Deletion",

JOptionPane.YES\_NO\_OPTION,

JOptionPane.WARNING\_MESSAGE

);

}

void deleteBook()

{

try

{

FileWriter fileWriter = new FileWriter(file);

BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);

for (Book b : books)

{

if(b.bookId == book.bookId) continue;

bufferedWriter.write(b.getBookId() + "\*" +

b.getBookName() + "\*" +

b.getAuthorNames() + "\*" +

b.getPublication() + "\*" +

b.getDateOfPublication() + "\*" +

b.getPriceOfBook() + "\*" +

b.getTotalQuantityToOrder());

bufferedWriter.newLine();

}

bufferedWriter.close();

fileWriter.close();

}

catch (Exception e)

{

e.printStackTrace();

System.out.println("Error writing to file!");

}

}

}