

Lab program 3 -

Create a class Book which contains four members : name, author, price, num-pages.

Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner ;  
class book {  
    private String name ;  
    private String author ;  
    private double price ;  
    private int numPages ;  
    public Book (String name, String author, double price,  
                int numPages) {  
        this.name = name ;  
        this.author = author ;  
        this.price = price ;  
        this.numPages = this.numPages ;  
    }  
    public void setName () {  
        this.name = name ;  
    }  
    public String getName () {  
        return name ;  
    }
```

```
public void setAuthor (String author) {  
    this.author = author; }  
public String getAuthor () {  
    return author; }  
public void setPrice (double price) {  
    this.price = price; }  
public double getPrice () {  
    return price; }  
public void setNumPages (int numPages) {  
    this.numPages = numPages; }  
public int getNumPages () {  
    return numPages; }  
public String toString () {  
    return "Book Details : \n Name : " + name +  
           "\n Author : " + author + "\n Price: INR "  
           + price + "\n Num of pages : " + numPages; }  
public class Main {  
    public static void main (String [] args) {  
        Scanner s = new Scanner (System.in);  
        System.out.println ("Enter the no. of books : ");  
        int n = s.nextInt ();  
        Book [] books = new Book [n];  
        for (int i=0; i<n; i++) {  
            books[i] = new Book ();  
            books[i].setAuthor (s.nextLine());  
            books[i].setPrice (s.nextDouble());  
            books[i].setNumPages (s.nextInt());  
        }  
        for (int i=0; i<n; i++) {  
            System.out.println (books[i]);  
        }  
    }  
}
```

System.out.println ("Enter name : ");

String name = s.nextLine();

System.out.println ("Enter author : ");

String author = s.nextLine();

System.out.println ("Enter Price : ");

double price = s.nextInt();

System.out.println ("Enter number of pages : ");

int numPages = s.nextInt();

books[i] = new Book (name, author, price, numPages);

System.out.println ("In book details ");

for (int i=0; i<n; i++) {

System.out.println ("In Book "+(i+1)+" : In
+book));

}

s.close();

}

}

Output —

Enter the number of books ::

Enter name :

dff + nmo +

Enter author :

dff

Enter price ::

800

Enter number of pages :

167

total = 167

avg = 5.00

use : 0 = 1 time load

Enter details for book 2 : Price : INR 800

Enter name : Name : kfdf Num of pages : 67

kfdf

Enter author :

abc

Book 2 :

Name : kfdf

Enter price :

700

Author : abc

Price : INR 700

Enter number of pages :

90

Num of pages : 90

Details of all books :

Book 1 :

Name : dff

Author : dff

lab pgm - 3

Algorithm → go to efficient programmatic writing with notes

Start —

Step 1 : start

Step 2 : Define the book class

Step 3 : declare members , name , author , price
and num pages

Step 4 : Add a toString method to display details
of book

Step 5 : Create book objects in main class

Step 6 : Initialize n , array book [] ; 1 qst2

Step 7 : Stop with working and robust : 0 qst2

() next taking books from user & adding
abstraction & encapsulation with - due : 1 qst2

• () next taking books from user & adding
abstraction & encapsulation with - due : 1 qst2

our statement , bracket next & returning : 2 qst2

min max : 1 qst2

book algorithm for - 2 : 1 qst2

• () next , return

implementation with others bracket mind : 1 qst2

• shiv's book algorithm

qst2 : 2 qst2

3.

Start

Enter number of books

initialise array of books[n]

Enter book details name, author, price

create new book obj

$i < n$

T

new book obj

End

Point details of all books