**CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING**



**Advanced Computing Training School**

**Course Name:** PG- Diploma in Advanced Computing

**Batch:** SEP 2021

**Module Name:** Java with oops **Date: 27/11/2021 Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Max Marks: 40 Marks**

**PRN No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Duration: 2 Hours**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Q1. (25 Marks)**

**Write a Book class with following properties:**

- Member variables

o String bookName

o double bookCost

o String authorName

- Overloaded constructors

o No arguments constructor

o Constructor that accepts bookName and bookCost(authorName is hardcoded)

o Constructor that accepts bookName, authorName and bookCost

- Override toString

- Other methods of own choice

**Write a menu driven program to demonstrate following operations using ArrayList of "Book":**

**1. Add Book to list**

**2. Remove Book list**

**3. Show all Books**

**4. Show all Books in sorted order (sort on the basis of bookCost)**

**5. Find a book with bookName**

**6. Save all Books details into file**

**7. Quit**

**import java.io.Serializable;**

**public class Book implements Serializable{**

**private String bookName,authorName;**

**private double bookCost;**

**public Book() {**

**//No argument constructor**

**}**

**public Book(String bookName, double bookCost) {**

**this.bookName = bookName;**

**this.bookCost = bookCost;**

**authorName="P. K. Warma";**

**}**

**public Book(String bookName, String authorName, double bookCost) {**

**this.bookName = bookName;**

**this.authorName = authorName;**

**this.bookCost = bookCost;**

**}**

**public String getBookName() {**

**return bookName;**

**}**

**public void setBookName(String bookName) {**

**this.bookName = bookName;**

**}**

**public String getAuthorName() {**

**return authorName;**

**}**

**public void setAuthorName(String authorName) {**

**this.authorName = authorName;**

**}**

**public double getBookCost() {**

**return bookCost;**

**}**

**public void setBookCost(double bookCost) {**

**this.bookCost = bookCost;**

**}**

**@Override**

**public String toString() {**

**return "Book [BookName=" + bookName + ", AuthorName=" + authorName + ", BookCost=" + bookCost + "]";**

**}**

**}//end of Book**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectOutputStream;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.Iterator;

import java.util.Scanner;

public class BookUser {

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

ArrayList<Book> book=new ArrayList<>();

Scanner sc=new Scanner(System.in);

while(true) {

menu();

System.out.print("Enter Your Choice : ");

int choice=sc.nextInt();

if(choice == 7) break;

switch(choice) {

case 1:

{

addBook(book,sc);

break;

}

case 2:

{

removeBook(book,sc);

break;

}

case 3:

{

showBooks(book);

break;

}

case 4:

{

Collections.sort(book,new SortByCost());

for(Book bk : book) {

System.out.println(bk);

}

break;

}

case 5:

{

findBook(book,sc);

break;

}

case 6:

{

saveBooksInFile(book);

break;

}

default : System.out.println("Enter valid choice");

}//end of switch

}//end of while

}//end of main

private static void saveBooksInFile(ArrayList<Book> book) throws IOException {

FileOutputStream fout= new FileOutputStream("G:\\Temp\\exam.book"); //create new file

ObjectOutputStream oos= new ObjectOutputStream(fout);

for(Book bk :book) {

oos.writeObject(bk); //write all objects to file

}

}

private static void findBook(ArrayList<Book> book, Scanner sc) {

System.out.print("Enter Book Name For Search : ");

String bookName=sc.next();

for(Book bk : book) {

if(bk.getBookName().equals(bookName)) System.out.println(bk);

}

}

private static void removeBook(ArrayList<Book> book , Scanner sc) {

System.out.print("Enter Book Name For Remove : ");

String bookName=sc.next();

for(int i=0;i<book.size();i++) {

if(book.get(i).getBookName().equals(bookName)) book.remove(i);

}

}

private static void showBooks(ArrayList<Book> book) {

Iterator<Book> itt=book.iterator();

while(itt.hasNext()) {

System.out.println(itt.next());

}

}//end of showBook

private static void addBook(ArrayList<Book> book, Scanner sc) {

System.out.print("Enter Book Name : ");

String bookName=sc.next();

System.out.print("Enter Author Name : ");

String authorName=sc.next();

System.out.print("Enter Book Cost : ");

int bookCost=sc.nextInt();

Book bk=new Book(bookName, authorName, bookCost);

book.add(bk);

}//end of addBook

private static void menu() {

System.out.println("=-=-=-=-=-=-=-=-=-=-=-=-=-=");

System.out.println("1. Add Book To List");

System.out.println("2. Remove Book From List");

System.out.println("3. Show All Books");

System.out.println("4. Show All Books Sorted On Cost");

System.out.println("5. Find Book By Book Name");

System.out.println("6. Save All Book Details In File");

System.out.println("7. Quit");

}//end of menu

}//end of BookUser

class SortByCost implements Comparator<Book>{

@Override

public int compare(Book o1, Book o2) {

if(o1.getBookCost()>o2.getBookCost()) return 1;

if(o1.getBookCost()<o2.getBookCost()) return -1;

else return 0;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Q2. (15 Marks)**

**Write a program that creates a HashMap of rollnumber and name**.

Accept 5 pairs of rollnumber and name from user and add to hashmap

Show all rollnumbers in the hashmap[ use Iterator ]

Show all names in the hashmap [ use for loop without index

import java.util.Collection;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Set;

public class MyHashMap {

public static void main(String[] args) {

HashMap<Integer,String> students=new HashMap<>();

students.put(1, "amey");

students.put(2, "abhi");

students.put(3, "jitu");

students.put(4, "rutuja");

students.put(5, "vanita");

Set<Integer> key=students.keySet();

System.out.println("All Roll No. In HashMap : ");

Iterator<Integer> itt=key.iterator();

while(itt.hasNext()) {

System.out.println(itt.next());

}

Collection<String> names =students.values();

System.out.println("All Names In HashMap : ");

for(String nm : names) {

System.out.println(nm);

}

}//end of main

}//end of class

**Evaluation of Lab Exam should be based on the following criteria**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Details** | **Max**  **Marks** | **Marks**  **Obtain** |
| **Algorithm** | **Documentation of Algorithm and Flowchart** | **2** |  |
| **Program adheres to the algorithm and flowchart** | **2** |  |
| **Efficiency** | **Program is using only the required number of variables**  **/conditions/loops/pointers etc and is optimal** | **2** |  |
| **Correctness** | **The program produces desired output for a given input** | **25**  **5** |  |
| **The program handles all valid and Invalid inputs** |  |
| **Software**  **Engineering**  **Principles** | **The program has meaning variable/function names** | **2** |  |
| **The program is commented properly (At least 20% of the**  **code should be commented)** | **2** |  |
|  | ***Total Marks*** | **40** |  |
| cccccccc |  |  |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Signature of Student** **Signature of Evaluator** **Signature of Coordinator**

**Page 1 of 1**