

LAB ASSIGNMENT

Binary File Handling in Python

1. Write a Python program that manages employee records using a binary file. The program should have the following functions:

add_employee()

This function should prompt the user to input data for an employee, including Employee ID, Name, Department, and Salary. The data should be stored in a dictionary format: {'ID': ID, 'Name': Name, 'Department': Department, 'Salary': Salary}. The function should then add the dictionary object to the emp.dat binary file using pickling.

display_records()

This function should display all the records of employees stored in emp.dat file on the screen. Each employee's details should be displayed with their corresponding attributes: Employee ID, Name, Department, and Salary.

search_employee(id)

This function should receive an Employee ID as an argument. It should search through the emp.dat file and display the details of the employee with the given ID. If no employee with the provided ID is found, display an appropriate message.

In the program, include a menu-based interface that allows users to choose which function to execute.

Sample Output:

```
Employee Record Management System
```

```
Menu:
```

- ```
1. Add Employee
2. Display Employee Records
3. Search Employee
4. Exit
```

```
Enter your choice: 1
```

```
Enter Employee ID: 101
Enter Employee Name: Rajesh Kumar
Enter Employee Department: IT
Enter Employee Salary: 75000
Employee added successfully.
```

```
Menu:
```

- ```
1. Add Employee
2. Display Employee Records
3. Search Employee
4. Exit
```

```
Enter your choice: 2
```

```
Employee Records:
Employee ID: 101
Name: Rajesh Kumar
Department: IT
Salary: 75000
-----
```

```
Menu:
1. Add Employee
2. Display Employee Records
3. Search Employee
4. Exit
```

Enter your choice: 3

```
Enter Employee ID to search: 101
Employee Details:
Employee ID: 101
Name: Rajesh Kumar
Department: IT
Salary: 75000
```

```
Menu:
1. Add Employee
2. Display Employee Records
3. Search Employee
4. Exit
```

Enter your choice: 3

```
Enter Employee ID to search: 102
Employee with ID 102 not found.
```

```
Menu:
1. Add Employee
2. Display Employee Records
3. Search Employee
4. Exit
```

Enter your choice: 4

Thanks for using Employee Record Management System

2. Write a Python program that manages book records using a binary file called book.dat. The program should have the following functions:

add_record()

This function should prompt the user to input data for a book, including the Book Number, Book Name, Author, and Price. The data should be stored in a list format: [BookNo, Book_Name, Author, Price]. The function should then add the book data to the book.dat binary file using pickling.

display_records()

This function should display all the records stored in the book.dat file on the screen. Each record should be displayed with its corresponding attributes: Book Number, Book Name, Author, and Price.

books_by_author()

This function should ask the user to input an author's name. It should then search through the records in the book.dat file and display all the records for books written by the given author.

books_by_price(price)

This function should receive a maximum price as an argument. It should search through the records in the book.dat file and display all the records for books with a price less than the provided maximum price.

copy_data()

This function should read the contents of the book.dat file. It should copy the records of books whose price is more than Rs. 500 to a new file named costly_book.dat using pickling. The function should return the count of records copied.

delete_record(book_number)

This function should receive book number as an argument. If a record with the given book number exists, it should be deleted from the book.dat file. If the record does not exist, display an error message.

update_record(book_number)

This function should receive book number as an argument. If a record with the given book number exists, prompt the user to update the Book Name, Author, and Price for that record. If the record does not exist, display an error message.

In the program, include a menu-based interface that allows users to choose which function to execute. Sample output of menu is:

Menu:

1. Add Record
2. Display Records
3. Display Books by Author
4. Display Books by Price
5. Copy Expensive Books
6. Delete Record
7. Update Record
8. Exit

Enter your choice: _