Lab Practice Set 4 - Structure Programming

Embedded programming

November 12, 2024

1. Design a Digital Library System with Multiple Books

Question: You are tasked with building a **Digital Library System** that manages information about books in a library. Define a structure `Book` with the following fields:

- `title` (string, up to 50 characters)
- `author` (string, up to 50 characters)
- `price` (float)

You need to create an array of 5 `Book` structures. The program should allow the user to input details for 5 books and then display all the details in a neat tabular format.

Expected Output:

```
Enter details for 5 books:

Book 1:
Title: The C Programming Language
Author: Brian W. Kernighan
Price: 499.99

Book 2:
Title: The Art of Computer Programming
Author: Donald Knuth
Price: 1299.99

Book 3:
Title: Clean Code
Author: Robert C. Martin
Price: 899.99

Book 4:
Title: Introduction to Algorithms
Author: Thomas H. Cormen
Price: 799.99

Book 5:
Title: The Pragmatic Programmer
Author: Andrew Hunt
Price: 699.99

Library Book List:
------------------------------------------------------------
| Title                          | Author               | Price   |
------------------------------------------------------------
| The C Programming Language     | Brian W. Kernighan   | 499.99  |
| The Art of Computer Programming| Donald Knuth         | 1299.99 |
| Clean Code                     | Robert C. Martin     | 899.99  |
```

```
| Introduction to Algorithms          | Thomas H. Cormen       | 799.99  |
| The Pragmatic Programmer            | Andrew Hunt            | 699.99  |
-------------------------------------------------------------
```

Topic Covered: - Array of structures - Looping through arrays - Displaying data in a tabular format

2. Student Database Management System with Multiple Entries

Question: You have been assigned to develop a **Student Database Management System** for a school. Define a structure `Student` that contains the following fields:

- `name` (string, up to 30 characters)
- `age` (integer)
- `marks` (float)

Create an array of 5 `Student` structures. Input the details for 5 students, then display them in a neat tabular format.

Expected Output:

```
Enter details for 5 students:

Student 1:
Name: Alice
Age: 20
Marks: 85.5

Student 2:
Name: Bob
Age: 21
Marks: 90.0

Student 3:
Name: Charlie
Age: 22
Marks: 78.5

Student 4:
Name: David
Age: 23
Marks: 92.0

Student 5:
Name: Eve
Age: 24
Marks: 88.0

Student Database:
-------------------------------------------------------------
| Name     | Age | Marks              |
-------------------------------------------------------------
| Alice    | 20  | 85.5               |
| Bob      | 21  | 90.0               |
| Charlie  | 22  | 78.5               |
| David    | 23  | 92.0               |
| Eve      | 24  | 88.0               |
-------------------------------------------------------------
```

Topic Covered: - Array of structures - Looping through structures - Displaying data in a tabular format

3. Rectangle Area Calculator for Multiple Rectangles

Question: You have been asked to create a **Shape Calculator** to compute the area of multiple rectangles. Define a structure `Rectangle` that contains:

- `length` (float)
- `width` (float)

Create an array of 5 `Rectangle` structures. Input the length and width for 5 rectangles and then compute and display their areas.

Expected Output:

```
Enter details for 5 rectangles:

Rectangle 1:
Length: 6.5
Width: 4.0

Rectangle 2:
Length: 8.0
Width: 3.5

Rectangle 3:
Length: 5.0
Width: 7.0

Rectangle 4:
Length: 9.5
Width: 2.0

Rectangle 5:
Length: 4.5
Width: 5.5

Rectangle Areas:
-------------------------------------------------
| Rectangle | Length | Width | Area              |
-------------------------------------------------
| 1         | 6.5    | 4.0   | 26.0              |
| 2         | 8.0    | 3.5   | 28.0              |
| 3         | 5.0    | 7.0   | 35.0              |
| 4         | 9.5    | 2.0   | 19.0              |
| 5         | 4.5    | 5.5   | 24.75             |
-------------------------------------------------
```

Topic Covered: - Array of structures - Passing structures to functions - Structure member access in functions

4. Employee Payroll System with Multiple Employees

Question: You are building an **Employee Payroll System** for a company. Define a structure `Employee` that contains:

- `name` (string, up to 30 characters)
- `id` (integer)
- `salary` (float)

Create an array of 5 `Employee` structures. Input the details of 5 employees and display the payroll information for each employee.

Expected Output:

```
Enter details for 5 employees:

Employee 1:
Name: James
ID: 101
Salary: 55000.75

Employee 2:
Name: Emma
ID: 102
Salary: 60000.50

Employee 3:
Name: Liam
ID: 103
Salary: 67000.80

Employee 4:
Name: Olivia
ID: 104
Salary: 72000.90

Employee 5:
Name: Sophia
ID: 105
Salary: 75000.00

Employee Payroll:
-----------------------------------------------------------
| Name    | ID   | Salary              |
-----------------------------------------------------------
| James   | 101  | 55000.75          |
| Emma    | 102  | 60000.50          |
| Liam    | 103  | 67000.80          |
| Olivia  | 104  | 72000.90          |
| Sophia  | 105  | 75000.00          |
-----------------------------------------------------------
```

Topic Covered: - Array of structures - Looping through arrays - Displaying data in a tabular format

5. Student Marks Update System with Multiple Entries

Question: In a **Student Marks Management System**, define a structure `Student` to store the following:

- `name` (string, up to 30 characters)
- `rollNo` (integer)
- `marks` (float)

Create an array of 5 `Student` structures. Input the details for 5 students, and then update the marks for each student. Finally, display the updated details.

Expected Output:

```
Enter details for 5 students:

Student 1:
Name: Alice
Roll Number: 101
Marks: 75.0

Student 2:
Name: Bob
Roll Number: 102
Marks: 80.0

Student 3:
Name: Charlie
Roll Number: 103
Marks: 90.0

Student 4:
Name: David
Roll Number: 104
Marks: 65.0

Student 5:
Name: Eve
Roll Number: 105
Marks: 70.0

Updated Marks for All Students:
-----------------------------------------------------------
| Name     | Roll No | Marks               |
-----------------------------------------------------------
| Alice    | 101     | 80.0                |
| Bob      | 102     | 85.0                |
| Charlie  | 103     | 95.0                |
| David    | 104     | 70.0                |
| Eve      | 105     | 75.0                |
-----------------------------------------------------------
```

Topic Covered: - Array of structures - Looping through arrays - Updating structure members

6. Student Report Generation System with Marks and Grade

Question: Create a **Student Report Generation System** where a structure `Student` contains:

- `name` (string, up to 30 characters)
- `marks1`, `marks2`, `marks3` (floats)
- `grade` (string, up to 2 characters)

The system should calculate the total and average marks for each student and assign a grade based on the following:

- If average marks $\geq$ 90, grade = "A"
- If average marks $\geq$ 75 and ¡ 90, grade = "B"
- If average marks $\geq$ 60 and ¡ 75, grade = "C"
- If average marks ¡ 60, grade = "D"

Create an array of 5 `Student` structures, input data for each student, and display their report.

Expected Output:

```
Enter details for 5 students:

Student 1:
Name: Alice
Marks 1: 85
Marks 2: 90
Marks 3: 92

Student 2:
Name: Bob
Marks 1: 75
Marks 2: 70
Marks 3: 80

Student 3:
Name: Charlie
Marks 1: 50
Marks 2: 60
Marks 3: 55

Student 4:
Name: David
Marks 1: 95
Marks 2: 98
Marks 3: 92

Student 5:
Name: Eve
Marks 1: 60
Marks 2: 65
Marks 3: 55

Student Report:
-------------------------------------------------------------
| Name     | Marks 1 | Marks 2 | Marks 3 | Average | Grade |
-------------------------------------------------------------
| Alice    | 85      | 90      | 92      | 89.0    | A     |
| Bob      | 75      | 70      | 80      | 75.0    | B     |
| Charlie  | 50      | 60      | 55      | 55.0    | D     |
| David    | 95      | 98      | 92      | 95.0    | A     |
| Eve      | 60      | 65      | 55      | 60.0    | C     |
-------------------------------------------------------------
```

Topic Covered: - Array of structures - Structure member calculations - Conditional statements (grading system)

7. Date and Time Management System

Question: Design a **Date and Time Management System** with a structure `DateTime` that contains the following fields:

- `day`, `month`, `year` (integer)
- `hour`, `minute`, `second` (integer)

Create an array of 5 `DateTime` structures to store and display the date and time for 5 events. The program should then display all the entered date-time values in a tabular format.

Expected Output:

```
Enter details for 5 events:

Event 1:
Day: 15
Month: 10
Year: 2023
Hour: 10
Minute: 45
Second: 30

Event 2:
Day: 12
Month: 5
Year: 2023
Hour: 8
Minute: 30
Second: 0

Event 3:
Day: 1
Month: 1
Year: 2024
Hour: 0
Minute: 0
Second: 0

Event 4:
Day: 20
Month: 7
Year: 2023
Hour: 14
Minute: 50
Second: 15

Event 5:
Day: 30
Month: 11
Year: 2024
Hour: 23
Minute: 59
Second: 59

Event Date and Time List:
-------------------------------------------------------------
| Event | Day | Month | Year | Hour | Minute | Second       |
-------------------------------------------------------------
| 1     | 15  | 10    | 2023 | 10   | 45     | 30           |
```

```
| 2     | 12  | 5     | 2023 | 8    | 30     | 0           |
| 3     | 1   | 1     | 2024 | 0    | 0      | 0           |
| 4     | 20  | 7     | 2023 | 14   | 50     | 15          |
| 5     | 30  | 11    | 2024 | 23   | 59     | 59          |
-------------------------------------------------------------
```

Topic Covered: - Array of structures - Structure with multiple data types (integers for day, time, etc.) - Displaying structured data in a formatted table

8. Vehicle Registration System

Question: Create a **Vehicle Registration System** where a structure `Vehicle` contains:

- `registrationNumber` (string, up to 20 characters)
- `model` (string, up to 20 characters)
- `ownerName` (string, up to 30 characters)
- `yearOfManufacture` (integer)

Create an array of 5 `Vehicle` structures, input details for 5 vehicles, and then display all the information in a tabular format.

Expected Output:

```
Enter details for 5 vehicles:

Vehicle 1:
Registration Number: AB1234CD
Model: Toyota Corolla
Owner Name: Alice
Year of Manufacture: 2015

Vehicle 2:
Registration Number: XY9876ZA
Model: Honda Civic
Owner Name: Bob
Year of Manufacture: 2020

Vehicle 3:
Registration Number: GP4321LM
Model: Ford Mustang
Owner Name: Charlie
Year of Manufacture: 2018

Vehicle 4:
Registration Number: ZX3456MN
Model: Tesla Model 3
Owner Name: David
Year of Manufacture: 2022

Vehicle 5:
Registration Number: JK6789OP
Model: BMW X5
Owner Name: Eve
Year of Manufacture: 2021

Vehicle Registration List:
-------------------------------------------------------------
| Registration Number | Model         | Owner    | Year of Manufacture |
```

```
-------------------------------------------------------------
| AB1234CD            | Toyota Corolla| Alice    | 2015              |
| XY9876ZA            | Honda Civic   | Bob      | 2020              |
| GP4321LM            | Ford Mustang  | Charlie  | 2018              |
| ZX3456MN            | Tesla Model 3 | David    | 2022              |
| JK6789OP            | BMW X5        | Eve      | 2021              |
-------------------------------------------------------------
```

Topic Covered: - Array of structures - Multiple string members in a structure - Tabular display of structured data

9. Course Registration System

Question: In a **Course Registration System**, define a structure `Course` that holds:

- `courseName` (string, up to 40 characters)
- `courseCode` (string, up to 10 characters)
- `creditHours` (integer)

Create an array of 5 `Course` structures. Allow users to input data for 5 courses, and then display the course list with their respective credit hours.

Expected Output:

```
Enter details for 5 courses:

Course 1:
Course Name: Data Structures
Course Code: CS101
Credit Hours: 3

Course 2:
Course Name: Algorithms
Course Code: CS102
Credit Hours: 4

Course 3:
Course Name: Operating Systems
Course Code: CS201
Credit Hours: 3

Course 4:
Course Name: Database Management
Course Code: CS202
Credit Hours: 4

Course 5:
Course Name: Software Engineering
Course Code: CS301
Credit Hours: 3

Course Registration List:
-------------------------------------------------------------
| Course Name              | Course Code | Credit Hours |
-------------------------------------------------------------
| Data Structures          | CS101       | 3            |
| Algorithms               | CS102       | 4            |
| Operating Systems        | CS201       | 3            |
| Database Management      | CS202       | 4            |
```

```
| Software Engineering     | CS301       | 3            |
-------------------------------------------------------------
```

Topic Covered: - Array of structures - String manipulation - Data input and display in tabular format

10. Employee Salary Management System

Question: Design a **Salary Management System** using a structure `Employee` that contains:

- `name` (string, up to 30 characters)
- `employeeID` (integer)
- `basicSalary` (float)
- `bonus` (float)
- `totalSalary` (float)

Write a program to input the details for 5 employees, calculate their total salary (basic salary + bonus), and display the employee details.

Expected Output:

```
Enter details for 5 employees:

Employee 1:
Name: Alice
Employee ID: 101
Basic Salary: 50000
Bonus: 5000

Employee 2:
Name: Bob
Employee ID: 102
Basic Salary: 45000
Bonus: 4000

Employee 3:
Name: Charlie
Employee ID: 103
Basic Salary: 60000
Bonus: 7000

Employee 4:
Name: David
Employee ID: 104
Basic Salary: 55000
Bonus: 5500

Employee 5:
Name: Eve
Employee ID: 105
Basic Salary: 52000
Bonus: 5200

Employee Salary Details:
-------------------------------------------------------------
| Name     | Employee ID | Basic Salary | Bonus | Total Salary |
-------------------------------------------------------------
| Alice    | 101         | 50000        | 5000  | 55000        |
```

```
| Bob      | 102         | 45000        | 4000  | 49000        |
| Charlie  | 103         | 60000        | 7000  | 67000        |
| David    | 104         | 55000        | 5500  | 60500        |
| Eve      | 105         | 52000        | 5200  | 57200        |
-------------------------------------------------------------
```

Topic Covered: - Array of structures - Structure member calculations - Displaying structured data in tabular format