Experiment 8

Employee Search Using an Array of Objects

Problem Statement

Write a Java program to store employee details including employee number, name, and salary, and search for an employee by employee number.

Source code:

```
import java.util.Scanner;

class Employee {
    int empNo;
    String name;
    double salary;

    Employee(int empNo, String name, double salary) {
        this.empNo = empNo;
        this.name = name;
        this.salary = salary;
    }

    void display() {
        System.out.println("Employee Number: " + empNo);
        System.out.println("Name: " + name);
        System.out.println("Salary: " + salary);
    }
}

public class EmployeeSearch {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of employees: ");
        int n = scanner.nextInt();
        Employee[] employees = new Employee[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter Employee Number: ");
            int empNo = scanner.nextInt();
            scanner.nextLine(); // Consume newline
            System.out.print("Enter Name: ");
            String name = scanner.nextLine();
            System.out.print("Enter Salary: ");
            double salary = scanner.nextDouble();
            employees[i] = new Employee(empNo, name, salary);
        }

        System.out.print("Enter Employee Number to search: ");
        int searchNo = scanner.nextInt();
        boolean found = false;

        for (Employee emp : employees) {
            if (emp.empNo == searchNo) {
                System.out.println("Employee Found:");
                emp.display();
                found = true;
                break;
            }
        }

        if (!found) {
            System.out.println("Employee not found.");
        }
    }
}
```

Output:

```
24mca39@mcaserver:~/javalab/assign1$ java EmployeeSearch
Enter number of employees: 3
Enter Employee Number: 101
Enter Name: Akshay
Enter Salary: 50000
Enter Employee Number: 102
Enter Name: Nandana
Enter Salary: 45000
Enter Employee Number: 103
Enter Name: Yash
Enter Salary: 10000
Enter Employee Number to search: 106
Employee not found.
24mca39@mcaserver:~/javalab/assign1$ java EmployeeSearch
Enter number of employees: 3
Enter Employee Number: 101
Enter Name: Akshay
Enter Salary: 50000
Enter Employee Number: 102
Enter Name: Nandana
Enter Salary: 45000
Enter Employee Number: 103
Enter Name: Yash
Enter Salary: 10000
Enter Employee Number to search: 102
Employee Found:
Employee Number: 102
Name: Nandana
Salary: 45000.0
```

Experiment 9

String Search in an Array

Problem Statement

Write a Java program to store ‘n‘ strings in an array. Search for a given string. If found, print its index; otherwise, display ”String not found.”

Source code:

```java
import java.util.Scanner;

public class StringSearch {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of strings: ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        String[] strings = new String[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter string " + (i + 1) + ": ");
            strings[i] = scanner.nextLine();
        }

        System.out.print("Enter string to search: ");
        String searchStr = scanner.nextLine();
        boolean found = false;

        for (int i = 0; i < n; i++) {
            if (strings[i].equals(searchStr)) {
                System.out.println("String found at index: " + i);
                found = true;
                break;
            }
        }

        if (!found) {
            System.out.println("String not found.");
        }
    }
}
```

Output:

```
24mca39@mcaserver:~/javalab/assign1$ java StringSearch
Enter number of strings: 3
Enter string 1: Hi
Enter string 2: Hello
Enter string 3: World!
Enter string to search: hello
String not found.
24mca39@mcaserver:~/javalab/assign1$ java StringSearch
Enter number of strings: 3
Enter string 1: Hi
Enter string 2: Hello
Enter string 3: World!
Enter string to search: Hello
String found at index: 1
```

Experiment 10
String Manipulations

Problem Statement
Write a Java program to perform various string manipulations, including finding the length, converting to uppercase and lowercase, extracting characters and substrings, and reversing the string.

Source code:

```java
import java.util.Scanner;

public class StringManipulation {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = scanner.nextLine();

        // Length of the string
        System.out.println("Length: " + str.length());

        // Convert to uppercase and lowercase
        System.out.println("Uppercase: " + str.toUpperCase());
        System.out.println("Lowercase: " + str.toLowerCase());

        // Extract a character and substring
        if (str.length() > 2) {
            System.out.println("First character: " + str.charAt(0));
            System.out.println("Substring (first 3 characters): " + str.substring(0, 3));
        }

        // Reverse the string
        String reversed = new StringBuilder(str).reverse().toString();
        System.out.println("Reversed: " + reversed);
    }
}
```

Output:

```
24mca39@mcaserver:~/javalab/assign1$ java StringManipulation
Enter a string: Nandana
Length: 7
Uppercase: NANDANA
Lowercase: nandana
First character: N
Substring (first 3 characters): Nan
Reversed: anadnaN
```

Experiment 11

Inheritance in Java

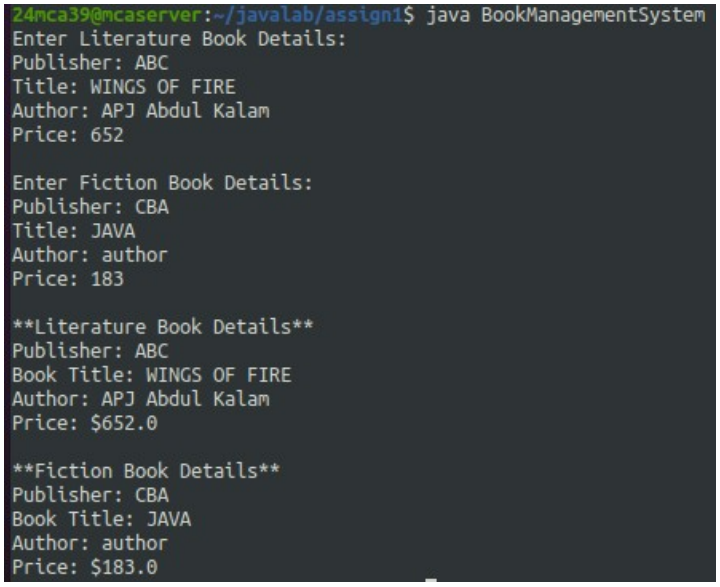
Problem Statement

Write a Java program to implement hierarchical inheritance for a book management system. Define a base class ‘Publisher‘, a derived class ‘Book‘, and two subclasses ‘Literature‘ and ‘Fiction‘. Include methods to read and display book details and demonstrate the functionality using user input.

Source code:

```java
import java.util.Scanner;

// Base class: Publisher
class Publisher {
    String publisherName;

    Publisher(String publisherName) {
        this.publisherName = publisherName;
    }

    void displayPublisher() {
        System.out.println("Publisher: " + publisherName);
    }
}

// Derived class: Book
class Book extends Publisher {
    String bookTitle;
    String author;
    double price;

    Book(String publisherName, String bookTitle, String author, double price) {
        super(publisherName);
        this.bookTitle = bookTitle;
        this.author = author;
        this.price = price;
    }

    void displayBookDetails() {
        displayPublisher();
        System.out.println("Book Title: " + bookTitle);
        System.out.println("Author: " + author);
        System.out.println("Price: $" + price);
    }
}

// Subclass: Literature
class Literature extends Book {
    Literature(String publisherName, String bookTitle, String author, double price) {
        super(publisherName, bookTitle, author, price);
    }

    void display() {
        System.out.println("\n**Literature Book Details**");
        displayBookDetails();
    }
}

// Subclass: Fiction
class Fiction extends Book {
    Fiction(String publisherName, String bookTitle, String author, double price) {
        super(publisherName, bookTitle, author, price);
    }

    void display() {
        System.out.println("\n**Fiction Book Details**");
        displayBookDetails();
    }
}

// Main class
public class BookManagementSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input for Literature Book
        System.out.println("Enter Literature Book Details:");
        System.out.print("Publisher: ");
        String litPublisher = scanner.nextLine();
        System.out.print("Title: ");
        String litTitle = scanner.nextLine();
        System.out.print("Author: ");
        String litAuthor = scanner.nextLine();
        System.out.print("Price: ");
        double litPrice = scanner.nextDouble();
        scanner.nextLine(); // Consume newline
        Literature literature = new Literature(litPublisher, litTitle, litAuthor, litPrice);

        // Input for Fiction Book
        System.out.println("\nEnter Fiction Book Details:");
        System.out.print("Publisher: ");
        String ficPublisher = scanner.nextLine();
        System.out.print("Title: ");
        String ficTitle = scanner.nextLine();
        System.out.print("Author: ");
        String ficAuthor = scanner.nextLine();
        System.out.print("Price: ");
        double ficPrice = scanner.nextDouble();
        Fiction fiction = new Fiction(ficPublisher, ficTitle, ficAuthor, ficPrice);

        // Display book details
        literature.display();
        fiction.display();
    }
}
```

Output:


24mca39@mcaserver:~/javalab/assign1$ java BookManagementSystem
Enter Literature Book Details:
Publisher: ABC
Title: WINGS OF FIRE
Author: APJ Abdul Kalam
Price: 652
Enter Fiction Book Details:
Publisher: CBA
Title: JAVA
Author: author
Price: 183
Literature Book Details
Publisher: ABC
Book Title: WINGS OF FIRE
Author: APJ Abdul Kalam
Price: $652.0
Fiction Book Details
Publisher: CBA
Book Title: JAVA
Author: author
Price: $183.0