Experiment 8

Aim

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;
import java.util.ArrayList;
class Employee{
        int empNumber;
        String name;
        double salary;
        public Employee(int empNumber, String name, double salary){
                this.empNumber = empNumber;
                this.name = name;
                this.salary = salary;
        }
        public static void main(String args[]){
                Scanner s = new Scanner(System.in);
                ArrayList<Employee> employees = new ArrayList<>();
                System.out.println("Enter the number of Employees: ");
                int n=s.nextInt();
                for (int i = 0; i < n; i++){
                        System.out.println("Enter details for Employee " + (i + 1) + ":");
                        System.out.print("Employee Number: ");
                        int empNumber = s.nextInt();
                        System.out.print("Employee Name: ");
                        String name = s.next();
                        System.out.print("Employee Salary: ");
                        double salary = s.nextDouble();
                        employees.add(new Employee(empNumber, name, salary));
                }
                System.out.print("\nEnter Employee Number to search: ");
                int searchEmpNumber = s.nextInt();
                boolean found = false;
                for (Employee emp : employees) {
                        if (emp.empNumber == searchEmpNumber) {
                                System.out.println("\nEmployee Found:");
                                System.out.println("Employee Number: " + emp.empNumber);
                                System.out.println("Employee Name: " + emp.name);
                                System.out.println("Employee Salary: " + emp.salary);
                                found = true;
                                break;
                        }
                }
                if (!found) System.out.println("Employee with Employee Number " + searchEmpNumber + " not found.");
    }
}
```

Output

```
24mca28@mcaserver:~/s2/ooplab/cycle-3$ java Employee
Enter the number of Employees:
3
Enter details for Employee 1:
Employee Number: 1
Employee Name: Akhil
Employee Salary: 45000
Enter details for Employee 2:
Employee Number: 2
Employee Name: Anshul
Employee Salary: 40000
Enter details for Employee 3:
Employee Number: 3
Employee Name: Ajin
Employee Salary: 50000

Enter Employee Number to search: 3

Employee Found:
Employee Number: 3
Employee Name: Ajin
Employee Salary: 50000.0
```

Experiment 9

Aim

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

```
import java.util.Scanner;
import java.util.ArrayList;
public class StringIndex{
        public static void main(String args[]){
                Scanner s = new Scanner(System.in);
                ArrayList<String> str=new ArrayList<>();
                System.out.println("Eneter the number of strings: ");
                int n = s.nextInt();
                System.out.println("Eneter the strings: ");
                for(int i = 0; i < n; i++)
                        str.add(s.next());
                System.out.println("Eneter the string to search: ");
                String key=s.next();
                boolean found=false;
                for(int i = 0; i < n; i++){
                        if(str.get(i).equals(key)){
                                System.out.println("Found at index: "+i);
                                found=true;
                                break;
                        }
                }
                if(!found) System.out.println("String not found!");
        }
}
```

Output

```
24mca28@mcaserver:~/s2/ooplab/cycle-3$ java StringIndex
Eneter the number of strings:
4
Eneter the strings:
aaaa
bbbb
cccc
dddd
Eneter the string to search:
cccc
Found at index: 2
```

Experiment 10

Aim

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

```
import java.util.Scanner;
public class StringManipulation{
        public static void main(String args[]){
                Scanner s=new Scanner(System.in);
                System.out.println("Enter a string: ");
                String str=s.next();
                System.out.println("length: "+str.length());
                System.out.println("uppercase: "+str.toUpperCase());
                System.out.println("lowercase: "+str.toLowerCase());
                System.out.println("Enter an index to extract: ");
                int i=s.nextInt();
                System.out.println("character at "+i+" is "+str.charAt(i));
                System.out.println("Enter a start and end index to substring: ");
                int start=s.nextInt();
                int end=s.nextInt();
                System.out.println("substring: "+str.substring(start,end));
        }
}
```

Output

```
24mca28@mcaserver:~/s2/ooplab/cycle-3$ java StringManipulation
Enter a string:
hello
lenngth: 5
uppercase: HELLO
lowercase: hello
Enter an index to extract:
1
charcter at 1 is e
Enter a start and end index to substring:
1       4
substring: ell
```

Experiment 11

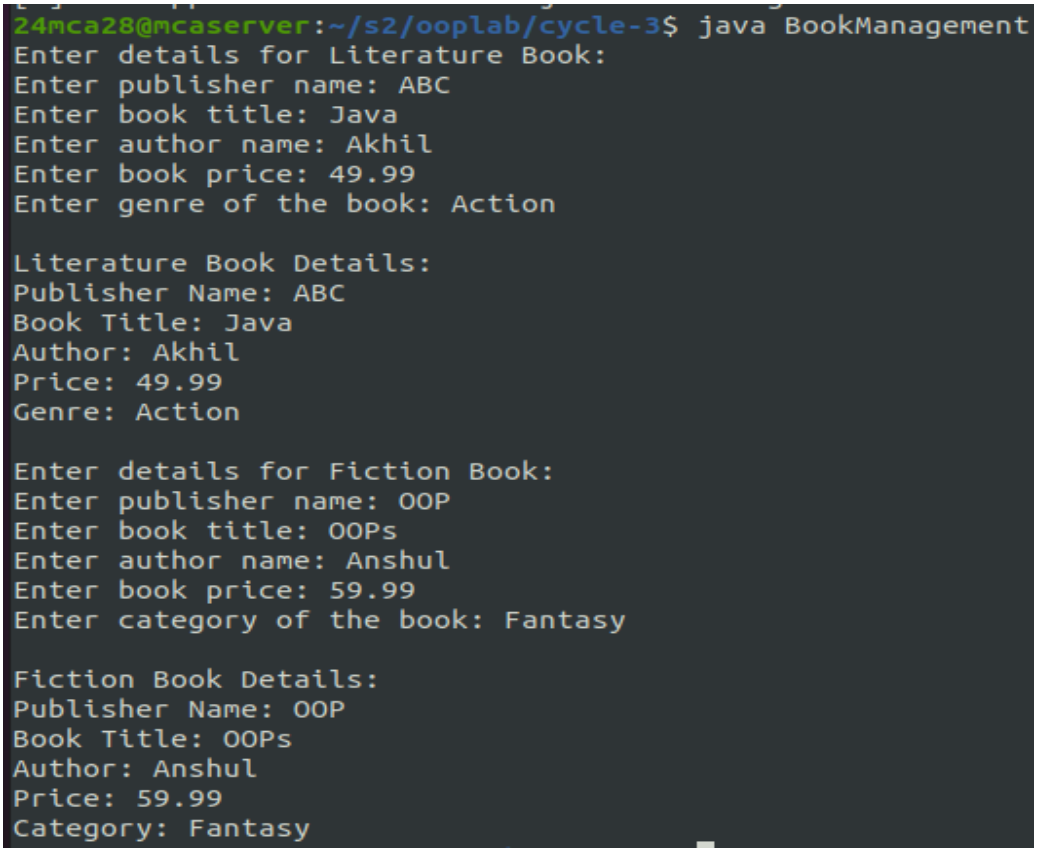
Aim

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

```
import java.util.Scanner;
class Publisher{
        String pname;
        public Publisher(String pname){
                this.pname = pname;
        }
        void display(){
                System.out.println("Publisher Name: " + pname);
        }
}
class Book extends Publisher{
        String title;
        String author;
        double price;
        public Book(String pname, String title, String author, double price) {
                super(pname);
                this.title = title;
                this.author = author;
                this.price = price;
        }
        void display(){
                super.display();
                System.out.println("Book Title: " + title);
                System.out.println("Author: " + author);
                System.out.println("Price: " + price);
        }
}
class Literature extends Book{
        String genre;
        public Literature(String pname, String title, String author, double price, String genre){
                super(pname, title, author, price);
                this.genre = genre;
        }
        void display(){
                super.display();
                System.out.println("Genre: " + genre);
        }
}
class Fiction extends Book{
        String category;
        public Fiction(String pname, String title, String author, double price, String category) {
                super(pname, title, author, price);
                this.category = category;
        }
        void display(){
                super.display();
                System.out.println("Category: " + category);
        }
}
public class BookManagement{
        public static void main(String args[]){
                Scanner s = new Scanner(System.in);
                System.out.println("Enter details for Literature Book:");
                System.out.print("Enter publisher name: ");
                String pname = s.next();
                System.out.print("Enter book title: ");
                String title = s.next();
                System.out.print("Enter author name: ");
                String author = s.next();
                System.out.print("Enter book price: ");
                double price = s.nextDouble();
                System.out.print("Enter genre of the book: ");
                String genre = s.next();
                Literature literatureBook = new Literature(pname, title, author, price, genre);
                System.out.println("\nLiterature Book Details:");
                literatureBook.display();
                System.out.println("\nEnter details for Fiction Book:");
                System.out.print("Enter publisher name: ");
                pname = s.next();
                System.out.print("Enter book title: ");
                title = s.next();
                System.out.print("Enter author name: ");
                author = s.next();
                System.out.print("Enter book price: ");
                price = s.nextDouble();
                System.out.print("Enter category of the book: ");
                String category = s.next();
                Fiction fictionBook = new Fiction(pname, title, author, price, category);
                System.out.println("\nFiction Book Details:");
                fictionBook.display();
        }
}
```

Output


24mca28@mcaserver:~/s2/ooplab/cycle-3$ java BookManagement
Enter details for Literature Book:
Enter publisher name: ABC
Enter book title: Java
Enter author name: Akhil
Enter book price: 49.99
Enter genre of the book: Action
Literature Book Details:
Publisher Name: ABC
Book Title: Java
Author: Akhil
Price: 49.99
Genre: Action
Enter details for Fiction Book:
Enter publisher name: OOP
Enter book title: OOPs
Enter author name: Anshul
Enter book price: 59.99
Enter category of the book: Fantasy
Fiction Book Details:
Publisher Name: OOP
Book Title: OOPs
Author: Anshul
Price: 59.99
Category: Fantasy