Aim : Develop Java programs using core concepts such as data structures, collections, and multithreading to manage and manipulate data.

Easy Level:
Write a Java program to implement an ArrayList that stores employee details (ID, Name, and Salary). Allow users to add, update, remove, and search employees.

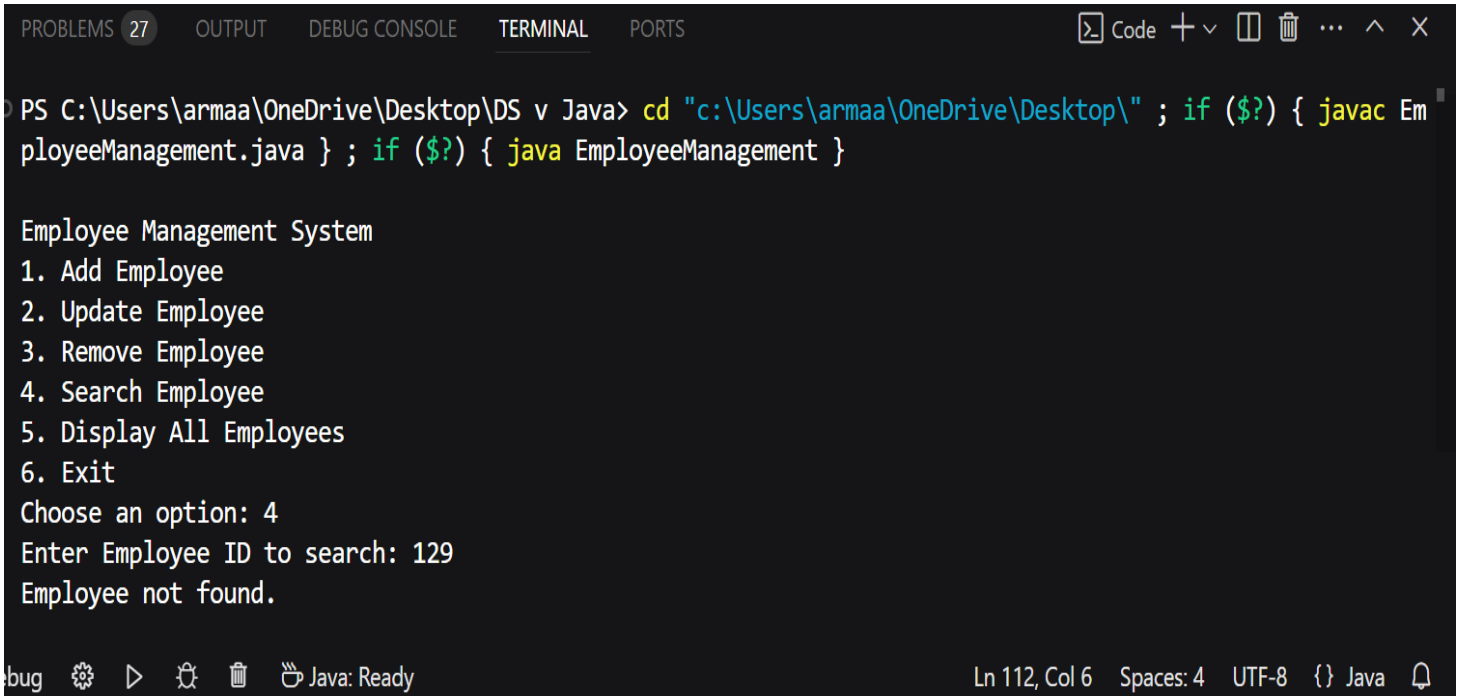
Code:

```
import java.util.*;

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

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

    @Override
    public String toString() {
        return "ID: " + id + ", Name: " + name + ", Salary: " + salary;
    }
}

public class EmployeeManagement {
    private static List<Employee> employeeList = new ArrayList<>();
    private static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        while (true) {
            System.out.println("\nEmployee Management System");
            System.out.println("1. Add Employee");
            System.out.println("2. Update Employee");
            System.out.println("3. Remove Employee");
            System.out.println("4. Search Employee");
            System.out.println("5. Display All Employees");
            System.out.println("6. Exit");
            System.out.print("Choose an option: ");
```

```
            int choice = scanner.nextInt();
            switch (choice) {
                case 1: addEmployee(); break;
                case 2: updateEmployee(); break;
                case 3: removeEmployee(); break;
                case 4: searchEmployee(); break;
                case 5: displayEmployees(); break;
                case 6: System.exit(0);
                default: System.out.println("Invalid choice. Try again.");
            }
        }
    }

    private static void addEmployee() {
        System.out.print("Enter Employee ID: ");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        System.out.print("Enter Employee Name: ");
        String name = scanner.nextLine();
        System.out.print("Enter Employee Salary: ");
        double salary = scanner.nextDouble();
        employeeList.add(new Employee(id, name, salary));
        System.out.println("Employee added successfully.");
    }

    private static void updateEmployee() {
        System.out.print("Enter Employee ID to update: ");
        int id = scanner.nextInt();
        for (Employee emp : employeeList) {
            if (emp.id == id) {
                scanner.nextLine(); // Consume newline
                System.out.print("Enter New Name: ");
                emp.name = scanner.nextLine();
                System.out.print("Enter New Salary: ");
                emp.salary = scanner.nextDouble();
                System.out.println("Employee updated successfully.");
                return;
            }
        }
        System.out.println("Employee not found.");
    }
```

```
    private static void removeEmployee() {
        System.out.print("Enter Employee ID to remove: ");
        int id = scanner.nextInt();
        Iterator<Employee> iterator = employeeList.iterator();
        while (iterator.hasNext()) {
            if (iterator.next().id == id) {
                iterator.remove();
                System.out.println("Employee removed successfully.");
                return;
            }
        }
        System.out.println("Employee not found.");
    }

    private static void searchEmployee() {
        System.out.print("Enter Employee ID to search: ");
        int id = scanner.nextInt();
        for (Employee emp : employeeList) {
            if (emp.id == id) {
                System.out.println("Employee Found: " + emp);
                return;
            }
        }
        System.out.println("Employee not found.");
    }

    private static void displayEmployees() {
        if (employeeList.isEmpty()) {
            System.out.println("No employees available.");
            return;
        }
        System.out.println("\nEmployee List:");
        for (Employee emp : employeeList) {
            System.out.println(emp);
        }
    }
}
```

Output:


PROBLEMS 27
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
PS C:\Users\armaa\OneDrive\Desktop\DS v Java> cd "c:\Users\armaa\OneDrive\Desktop\" ; if ($?) { javac EmployeeManagement.java } ; if ($?) { java EmployeeManagement }
Employee Management System
1. Add Employee
2. Update Employee
3. Remove Employee
4. Search Employee
5. Display All Employees
6. Exit
Choose an option: 4
Enter Employee ID to search: 129
Employee not found.
Java: Ready
Ln 112, Col 6
Spaces: 4
UTF-8
Java