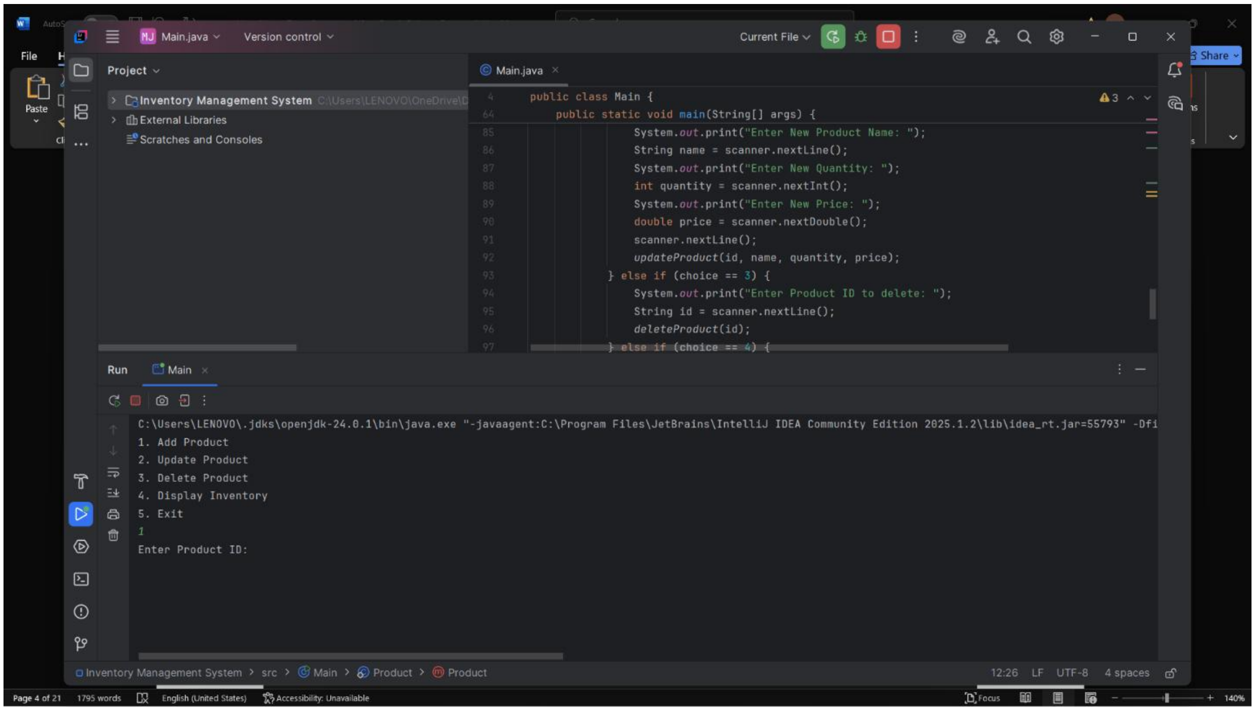
DATA STRUCTURE AND ALGORITHM:

Exercise 1: Inventory Management System

```
import java.util.HashMap;
import java.util.Scanner;

public class Main {
    static class Product {
        private String productId;
        private String productName;
        private int quantity;
        private double price;

        public Product(String productId, String productName, int quantity, double price) {
            this.productId = productId;
            this.productName = productName;
            this.quantity = quantity;
            this.price = price;
        }

        public String getProductId() {
            return productId;
        }

        public void setProductName(String productName) {
            this.productName = productName;
        }

        public void setQuantity(int quantity) {
            this.quantity = quantity;
        }

        public void setPrice(double price) {
            this.price = price;
    }

    public String toString() {
        return "ID: " + productId + ", Name: " + productName + ", Quantity: " + quantity + ", Price: " + price;
    }
}

private static HashMap<String, Product> inventory = new HashMap<>();

public static void addProduct(Product product) {
    inventory.put(product.getProductId(), product);
}

public static void updateProduct(String productId, String name, int quantity, double price) {
    if (inventory.containsKey(productId)) {
        Product product = inventory.get(productId);
        product.setProductName(name);
        product.setQuantity(quantity);
        product.setPrice(price);
    }
}

public static void deleteProduct(String productId) {
    inventory.remove(productId);
}

public static void displayInventory() {
    for (Product product : inventory.values()) {
        System.out.println(product);
    }
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    while (true) {
        System.out.println("1. Add Product\n2. Update Product\n3. Delete Product\n4. Display
Inventory\n5. Exit");
        int choice = scanner.nextInt();
        scanner.nextLine();
        if (choice == 1) {
            System.out.print("Enter Product ID: ");
            String id = scanner.nextLine();
            System.out.print("Enter Product Name: ");
            String name = scanner.nextLine();
            System.out.print("Enter Quantity: ");
            int quantity = scanner.nextInt();
            System.out.print("Enter Price: ");
            double price = scanner.nextDouble();
            scanner.nextLine();
            Product product = new Product(id, name, quantity, price);
            addProduct(product);
        } else if (choice == 2) {
            System.out.print("Enter Product ID to update: ");
            String id = scanner.nextLine();
            System.out.print("Enter New Product Name: ");
            String name = scanner.nextLine();
            System.out.print("Enter New Quantity: ");
            int quantity = scanner.nextInt();
            System.out.print("Enter New Price: ");
            double price = scanner.nextDouble();
            scanner.nextLine();
            updateProduct(id, name, quantity, price);
        } else if (choice == 3) {
            System.out.print("Enter Product ID to delete: ");
            String id = scanner.nextLine();
            deleteProduct(id);
        } else if (choice == 4) {
            displayInventory();
        } else if (choice == 5) {
            break;
            }
        }
        scanner.close();
    }
}
```

Output:



Main.java
Version control
Current File
Project
Inventory Management System C:\Users\LENOVO\OneDrive\D
External Libraries
Scratches and Consoles
Run
Main
Inventory Management System > src > Main > Product > Product
12:26 LF UTF-8 4 spaces
Page 4 of 21 1795 words English (United States) Accessibility: Unavailable
Focus
140%

```
public class Main {
    public static void main(String[] args) {
                System.out.print("Enter New Product Name: ");
                String name = scanner.nextLine();
                System.out.print("Enter New Quantity: ");
                int quantity = scanner.nextInt();
                System.out.print("Enter New Price: ");
                double price = scanner.nextDouble();
                scanner.nextLine();
                updateProduct(id, name, quantity, price);
            } else if (choice == 3) {
                System.out.print("Enter Product ID to delete: ");
                String id = scanner.nextLine();
                deleteProduct(id);
            } else if (choice == 4) {
```

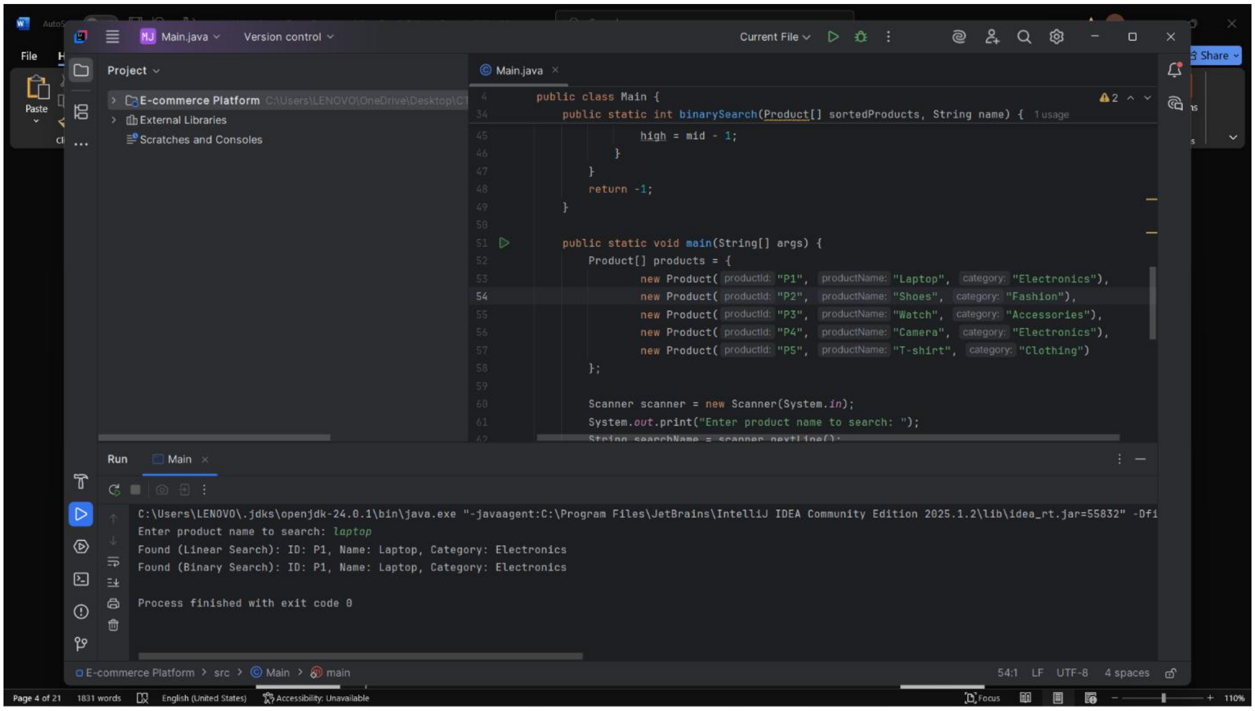
```
C:\Users\LENOVO\.jdks\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.2\lib\idea_rt.jar=55793" -Dfi
1. Add Product
2. Update Product
3. Delete Product
4. Display Inventory
5. Exit
1
Enter Product ID:
```

Exercise 2: E-commerce Platform Search Function

```
import java.util.Arrays;
import java.util.Scanner;

public class Main {
    static class Product implements Comparable<Product> {
        String productId;
        String productName;
        String category;
        public Product(String productId, String productName, String category) {
            this.productId = productId;
            this.productName = productName;
            this.category = category;
        }

        public int compareTo(Product other) {
            return this.productName.compareTo(other.productName);
        }

        public String toString() {
            return "ID: " + productId + ", Name: " + productName + ", Category: " + category;
        }
    }

    public static int linearSearch(Product[] products, String name) {
        for (int i = 0; i < products.length; i++) {
            if (products[i].productName.equalsIgnoreCase(name)) {
                return i;
            }
        }
        return -1;
    }

    public static int binarySearch(Product[] sortedProducts, String name) {
        int low = 0;
        int high = sortedProducts.length - 1;
        while (low <= high) {
            int mid = (low + high) / 2;
            int cmp = sortedProducts[mid].productName.compareToIgnoreCase(name);
            if (cmp == 0) {
                return mid;
            } else if (cmp < 0) {
                low = mid + 1;
            } else {
                high = mid - 1;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        Product[] products = {
                new Product("P1", "Laptop", "Electronics"),
                new Product("P2", "Shoes", "Fashion"),
                new Product("P3", "Watch", "Accessories"),
                new Product("P4", "Camera", "Electronics"),
                new Product("P5", "T-shirt", "Clothing")
        };

        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter product name to search: ");
        String searchName = scanner.nextLine();

        int linearIndex = linearSearch(products, searchName);
        if (linearIndex != -1) {
            System.out.println("Found (Linear Search): " + products[linearIndex]);
        } else {
            System.out.println("Product not found (Linear Search)");
        }

        Product[] sortedProducts = Arrays.copyOf(products, products.length);
        Arrays.sort(sortedProducts);
        int binaryIndex = binarySearch(sortedProducts, searchName);
        if (binaryIndex != -1) {
            System.out.println("Found (Binary Search): " + sortedProducts[binaryIndex]);
        } else {
            System.out.println("Product not found (Binary Search)");
        }
```

```
        scanner.close();
    }
}
```

Output:


Main.java
Version control
Current File
Project
E-commerce Platform C:\Users\LENOVO\OneDrive\Desktop\C
External Libraries
Scratches and Consoles
Main.java
Run
Main
E-commerce Platform > src > Main > main
54:1 LF UTF-8 4 spaces
Page 4 of 21 1831 words English (United States) Accessibility: Unavailable
Focus
110%

```
public class Main {
    public static int binarySearch(Product[] sortedProducts, String name) {  1 usage
                high = mid - 1;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        Product[] products = {
                new Product( productId: "P1",  productName: "Laptop",  category: "Electronics"),
                new Product( productId: "P2",  productName: "Shoes",  category: "Fashion"),
                new Product( productId: "P3",  productName: "Watch",  category: "Accessories"),
                new Product( productId: "P4",  productName: "Camera",  category: "Electronics"),
                new Product( productId: "P5",  productName: "T-shirt",  category: "Clothing")
        };

        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter product name to search: ");
```

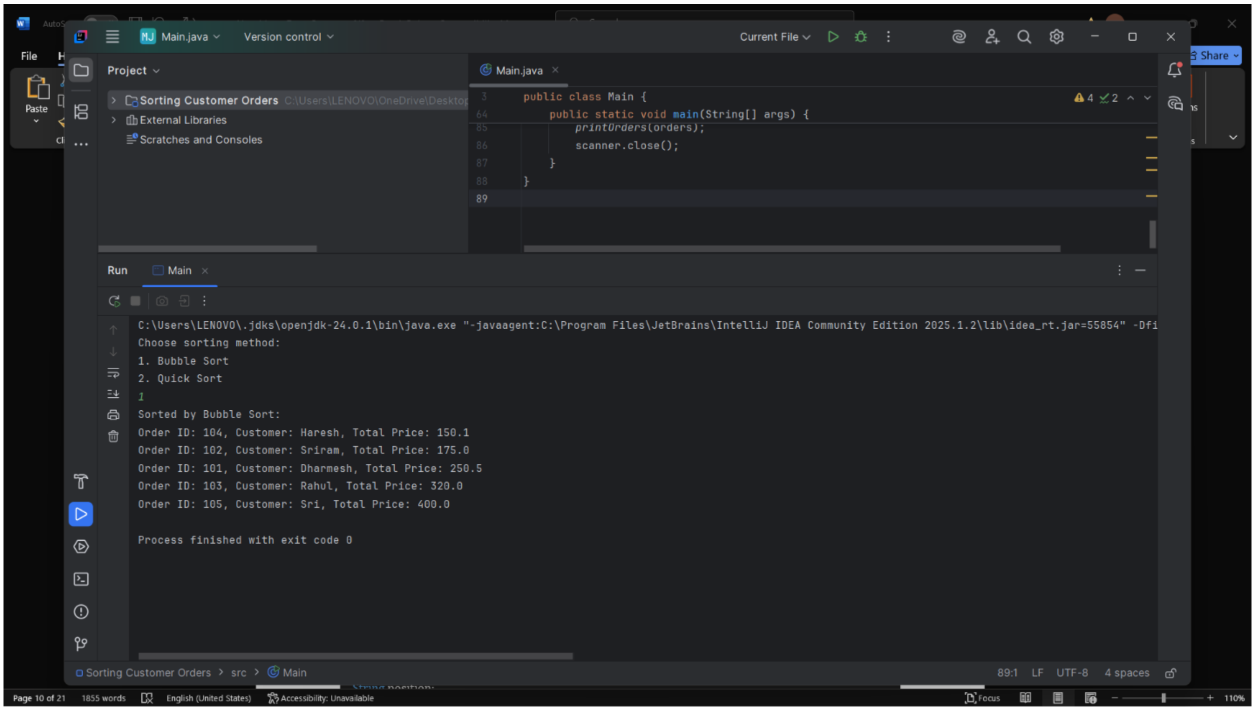
```
C:\Users\LENOVO\.jdks\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.2\lib\idea_rt.jar=55832" -Dfi
Enter product name to search: laptop
Found (Linear Search): ID: P1, Name: Laptop, Category: Electronics
Found (Binary Search): ID: P1, Name: Laptop, Category: Electronics

Process finished with exit code 0
```

Exercise 3: Sorting Customer Orders

```
import java.util.Scanner;

public class Main {
    static class Order {
        String orderId;
        String customerName;
        double totalPrice;

        public Order(String orderId, String customerName, double totalPrice) {
            this.orderId = orderId;
            this.customerName = customerName;
        this.totalPrice = totalPrice;
    }

    public String toString() {
        return "Order ID: " + orderId + ", Customer: " + customerName + ", Total Price: " + totalPrice;
    }
}

public static void bubbleSort(Order[] orders) {
    int n = orders.length;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (orders[j].totalPrice > orders[j + 1].totalPrice) {
                Order temp = orders[j];
                orders[j] = orders[j + 1];
                orders[j + 1] = temp;
            }
        }
    }
}

public static void quickSort(Order[] orders, int low, int high) {
    if (low < high) {
        int pi = partition(orders, low, high);
        quickSort(orders, low, pi - 1);
        quickSort(orders, pi + 1, high);
    }
}

public static int partition(Order[] orders, int low, int high) {
    double pivot = orders[high].totalPrice;
    int i = low - 1;
    for (int j = low; j < high; j++) {
        if (orders[j].totalPrice < pivot) {
            i++;
            Order temp = orders[i];
            orders[i] = orders[j];
            orders[j] = temp;
        }
    }
    Order temp = orders[i + 1];
    orders[i + 1] = orders[high];
    orders[high] = temp;
    return i + 1;
}

public static void printOrders(Order[] orders) {
    for (Order order : orders) {
        System.out.println(order);
    }
}

public static void main(String[] args) {
    Order[] orders = {
            new Order("101", "Dharmesh", 250.50),
            new Order("102", "Sriram", 175.00),
            new Order("103", "Rahul", 320.00),
            new Order("104", "Haresh", 150.10),
            new Order("105", "Sri", 400.00)
    };

    Scanner scanner = new Scanner(System.in);
    System.out.println("Choose sorting method:\n1. Bubble Sort\n2. Quick Sort");
    int choice = scanner.nextInt();

    if (choice == 1) {
        bubbleSort(orders);
        System.out.println("Sorted by Bubble Sort:");
    } else if (choice == 2) {
        quickSort(orders, 0, orders.length - 1);
```

```
            System.out.println("Sorted by Quick Sort:");
        }

        printOrders(orders);
        scanner.close();
    }
}
```

Output:



```
public class Main {
    public static void main(String[] args) {
        printOrders(orders);
        scanner.close();
    }
}
```

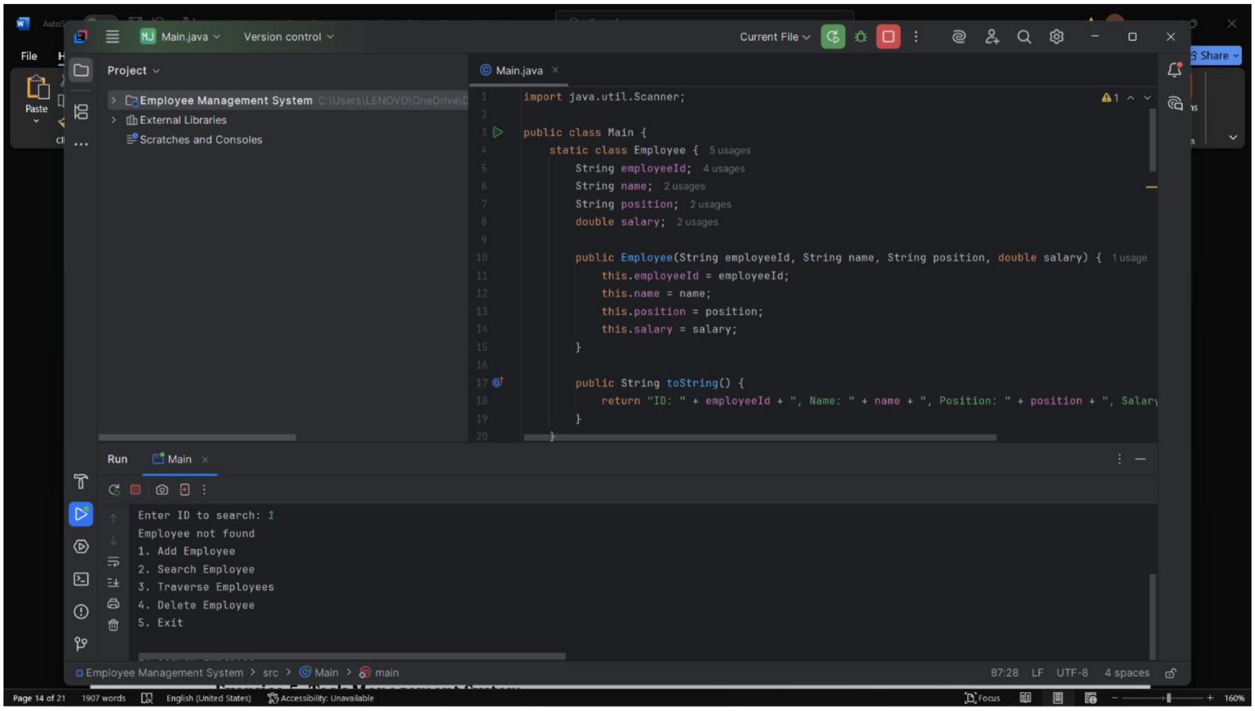
```
C:\Users\LENOVO\.jdks\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.2\lib\idea_rt.jar=55854" -Dfi
Choose sorting method:
1. Bubble Sort
2. Quick Sort
1
Sorted by Bubble Sort:
Order ID: 104, Customer: Haresh, Total Price: 150.1
Order ID: 102, Customer: Sriram, Total Price: 175.0
Order ID: 101, Customer: Dharmesh, Total Price: 250.5
Order ID: 103, Customer: Rahul, Total Price: 320.0
Order ID: 105, Customer: Sri, Total Price: 400.0

Process finished with exit code 0
```

Exercise 4: Employee Management System

```
import java.util.Scanner;

public class Main {
    static class Employee {
        String employeeId;
    String name;
    String position;
    double salary;

    public Employee(String employeeId, String name, String position, double salary) {
        this.employeeId = employeeId;
        this.name = name;
        this.position = position;
        this.salary = salary;
    }

    public String toString() {
        return "ID: " + employeeId + ", Name: " + name + ", Position: " + position + ", Salary: " + salary;
    }
}

static Employee[] employees = new Employee[100];
static int count = 0;

public static void addEmployee(Employee emp) {
    if (count < employees.length) {
        employees[count++] = emp;
    }
}

public static void searchEmployee(String id) {
    for (int i = 0; i < count; i++) {
        if (employees[i].employeeId.equals(id)) {
            System.out.println("Employee Found: " + employees[i]);
            return;
        }
    }
    System.out.println("Employee not found");
}
```

```
    public static void traverseEmployees() {
        for (int i = 0; i < count; i++) {
            System.out.println(employees[i]);
        }
    }

    public static void deleteEmployee(String id) {
        for (int i = 0; i < count; i++) {
            if (employees[i].employeeId.equals(id)) {
                for (int j = i; j < count - 1; j++) {
                    employees[j] = employees[j + 1];
                }
                employees[--count] = null;
                System.out.println("Employee deleted");
                return;
            }
        }
        System.out.println("Employee not found");
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("1. Add Employee\n2. Search Employee\n3. Traverse Employees\n4. Delete Employee\n5. Exit");
            int choice = scanner.nextInt();
            scanner.nextLine();
            if (choice == 1) {
                System.out.print("Enter ID: ");
                String id = scanner.nextLine();
                System.out.print("Enter Name: ");
                String name = scanner.nextLine();
                System.out.print("Enter Position: ");
                String position = scanner.nextLine();
                System.out.print("Enter Salary: ");
            double salary = scanner.nextDouble();
            scanner.nextLine();
            Employee emp = new Employee(id, name, position, salary);
            addEmployee(emp);
        } else if (choice == 2) {
            System.out.print("Enter ID to search: ");
            String id = scanner.nextLine();
            searchEmployee(id);
        } else if (choice == 3) {
            traverseEmployees();
        } else if (choice == 4) {
            System.out.print("Enter ID to delete: ");
            String id = scanner.nextLine();
            deleteEmployee(id);
        } else if (choice == 5) {
            break;
        }
    }
    scanner.close();
}
```

Output:



```
import java.util.Scanner;

public class Main {
    static class Employee {
        String employeeId;
        String name;
        String position;
        double salary;

        public Employee(String employeeId, String name, String position, double salary) {
            this.employeeId = employeeId;
            this.name = name;
            this.position = position;
            this.salary = salary;
        }

        public String toString() {
            return "ID: " + employeeId + ", Name: " + name + ", Position: " + position + ", Salary
        }
}
```

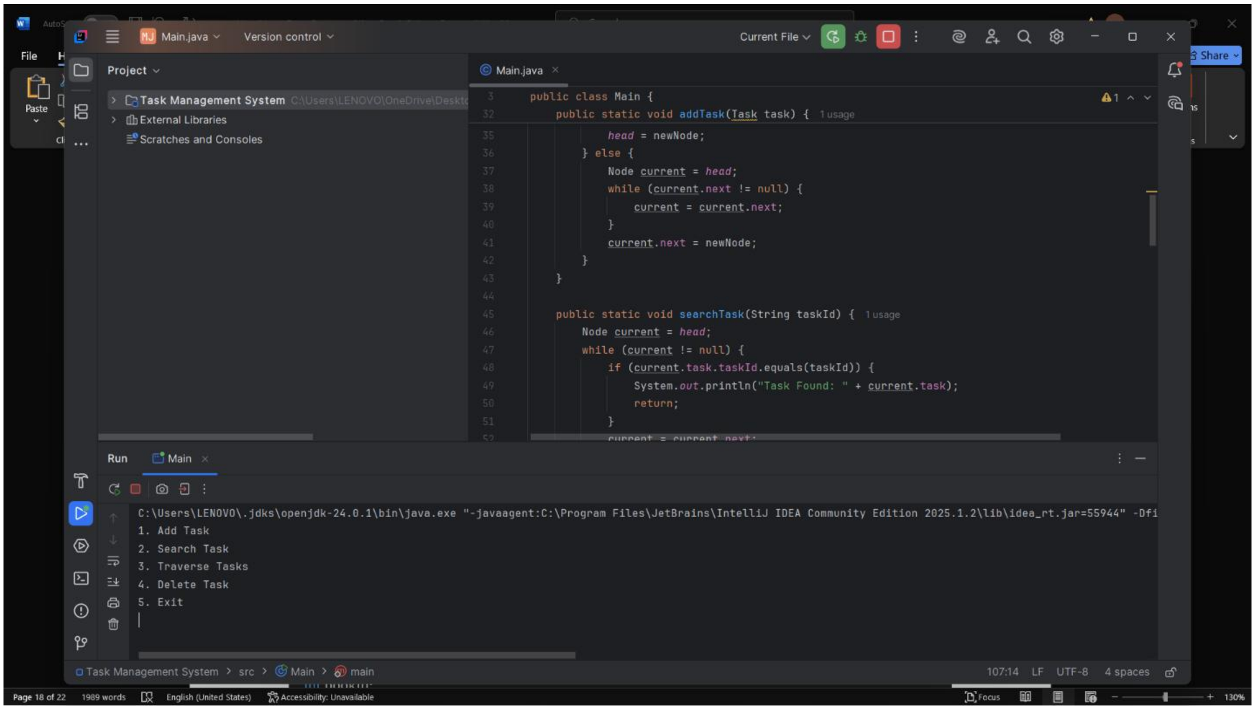
```
Enter ID to search: 1
Employee not found
1. Add Employee
2. Search Employee
3. Traverse Employees
4. Delete Employee
5. Exit
```

Exercise 5: Task Management System

```
import java.util.Scanner;

public class Main {
    static class Task {
        String taskId;
        String taskName;
        String status;

        public Task(String taskId, String taskName, String status) {
            this.taskId = taskId;
            this.taskName = taskName;
            this.status = status;
        }

        public String toString() {
            return "Task ID: " + taskId + ", Name: " + taskName + ", Status: " + status;
        }
```

```
}

static class Node {
    Task task;
    Node next;

    public Node(Task task) {
        this.task = task;
        this.next = null;
    }
}

static Node head = null;

public static void addTask(Task task) {
    Node newNode = new Node(task);
    if (head == null) {
        head = newNode;
    } else {
        Node current = head;
        while (current.next != null) {
            current = current.next;
        }
        current.next = newNode;
    }
}

public static void searchTask(String taskId) {
    Node current = head;
    while (current != null) {
        if (current.task.taskId.equals(taskId)) {
            System.out.println("Task Found: " + current.task);
            return;
        }
        current = current.next;
    }
    System.out.println("Task not found");
}

public static void traverseTasks() {
    Node current = head;
    while (current != null) {
        System.out.println(current.task);
        current = current.next;
    }
}

public static void deleteTask(String taskId) {
    if (head == null) {
        System.out.println("List is empty");
        return;
    }
    if (head.task.taskId.equals(taskId)) {
        head = head.next;
        System.out.println("Task deleted");
        return;
    }
    Node current = head;
    while (current.next != null) {
        if (current.next.task.taskId.equals(taskId)) {
            current.next = current.next.next;
            System.out.println("Task deleted");
            return;
        }
        current = current.next;
    }
    System.out.println("Task not found");
}

public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("1. Add Task\n2. Search Task\n3. Traverse Tasks\n4. Delete Task\n5. Exit");
            int choice = scanner.nextInt();
            scanner.nextLine();
            if (choice == 1) {
                System.out.print("Enter Task ID: ");
                String id = scanner.nextLine();
                System.out.print("Enter Task Name: ");
                String name = scanner.nextLine();
                System.out.print("Enter Status: ");
                String status = scanner.nextLine();
                addTask(new Task(id, name, status));
            } else if (choice == 2) {
                System.out.print("Enter Task ID to search: ");
                String id = scanner.nextLine();
                searchTask(id);
            } else if (choice == 3) {
                traverseTasks();
            } else if (choice == 4) {
                System.out.print("Enter Task ID to delete: ");
                String id = scanner.nextLine();
                deleteTask(id);
            } else if (choice == 5) {
                break;
            }
        }
        scanner.close();
    }
}
```

Output:



Main.java
Version control
Current File
Project
Task Management System C:\Users\LENOVO\OneDrive\Deskto
External Libraries
Scratches and Consoles
Run
Main
Task Management System > src > Main > main
107:14 LF UTF-8 4 spaces
Page 18 of 22 1989 words English (United States) Accessibility: Unavailable
Focus
130%

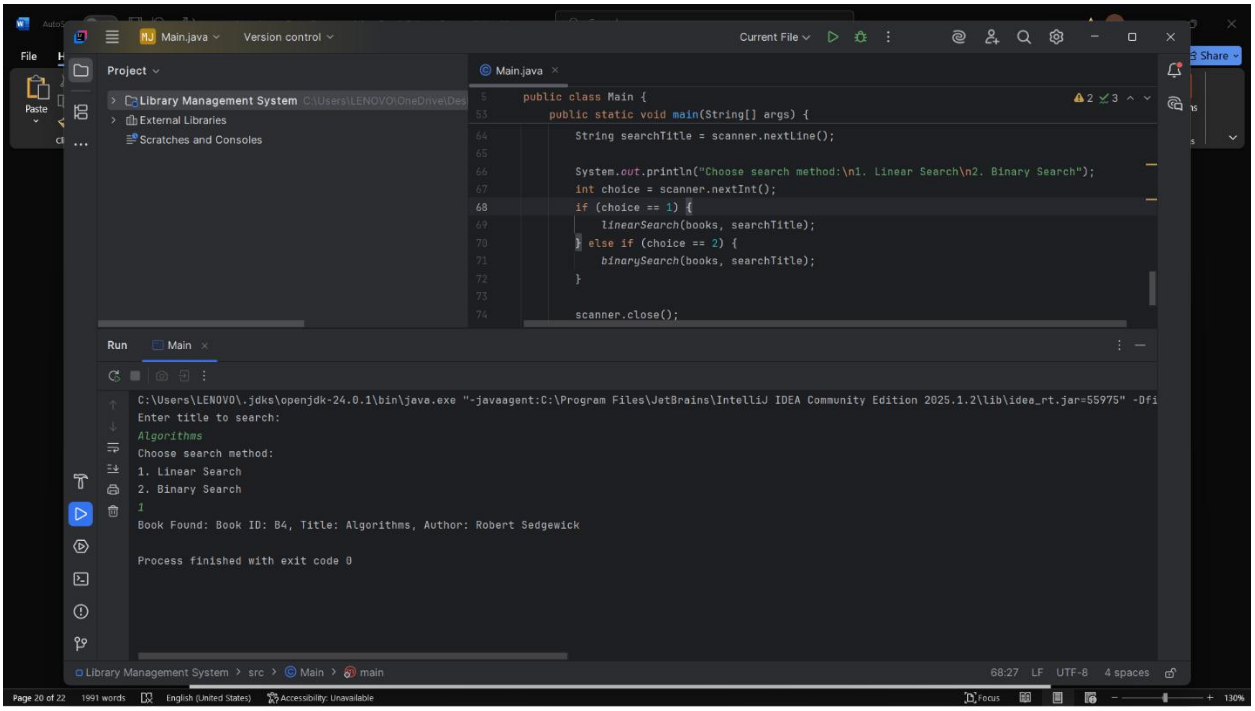
```java
public class Main {
    public static void addTask(Task task) {  1 usage
                head = newNode;
            } else {
                Node current = head;
                while (current.next != null) {
                    current = current.next;
                }
                current.next = newNode;
            }
        }

    public static void searchTask(String taskId) {  1 usage
        Node current = head;
        while (current != null) {
            if (current.task.taskId.equals(taskId)) {
                System.out.println("Task Found: " + current.task);
                return;
            }
            current = current.next;
```

```
C:\Users\LENOVO\.jdks\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.2\lib\idea_rt.jar=55944" -Dfi
1. Add Task
2. Search Task
3. Traverse Tasks
4. Delete Task
5. Exit
```

Exercise 6: Library Management System

```java
import java.util.Scanner;
import java.util.Arrays;
import java.util.Comparator;

public class Main {
  static class Book {
    String bookId;
    String title;
    String author;

    public Book(String bookId, String title, String author) {
      this.bookId = bookId;
      this.title = title;
      this.author = author;
    }

    public String toString() {
      return "Book ID: " + bookId + ", Title: " + title + ", Author: " + author;
    }
}

public static void linearSearch(Book[] books, String title) {
    boolean found = false;
    for (Book book : books) {
        if (book.title.equalsIgnoreCase(title)) {
            System.out.println("Book Found: " + book);
            found = true;
        }
    }
    if (!found) {
        System.out.println("Book not found");
    }
}

public static void binarySearch(Book[] books, String title) {
    Arrays.sort(books, Comparator.comparing(b -> b.title.toLowerCase()));
    int low = 0, high = books.length - 1;
    while (low <= high) {
        int mid = (low + high) / 2;
        int cmp = books[mid].title.compareToIgnoreCase(title);
        if (cmp == 0) {
            System.out.println("Book Found: " + books[mid]);
            return;
        } else if (cmp < 0) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }
    System.out.println("Book not found");
}

public static void main(String[] args) {
    Book[] books = {
        new Book("B1", "Data Structures", "Mark Allen"),
        new Book("B2", "Operating Systems", "Abraham Silberschatz"),
        new Book("B3", "Java Programming", "Herbert Schildt"),
        new Book("B4", "Algorithms", "Robert Sedgewick"),
        new Book("B5", "Computer Networks", "Andrew Tanenbaum")
    };

    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter title to search: ");
    String searchTitle = scanner.nextLine();

    System.out.println("Choose search method:\n1. Linear Search\n2. Binary Search");
    int choice = scanner.nextInt();
    if (choice == 1) {
      linearSearch(books, searchTitle);
    } else if (choice == 2) {
      binarySearch(books, searchTitle);
    }

    scanner.close();
  }
}
```

Output:


Main.java
Version control
Current File
Project
Library Management System C:\Users\LENOVO\OneDrive\Des
External Libraries
Scratches and Consoles
Run
Main
Library Management System > src > Main > main
68:27 LF UTF-8 4 spaces
Page 20 of 22 1991 words English (United States) Accessibility: Unavailable
Focus 130%

```
public class Main {
    public static void main(String[] args) {
        String searchTitle = scanner.nextLine();

        System.out.println("Choose search method:\n1. Linear Search\n2. Binary Search");
        int choice = scanner.nextInt();
        if (choice == 1) {
            linearSearch(books, searchTitle);
        } else if (choice == 2) {
            binarySearch(books, searchTitle);
        }

        scanner.close();
```

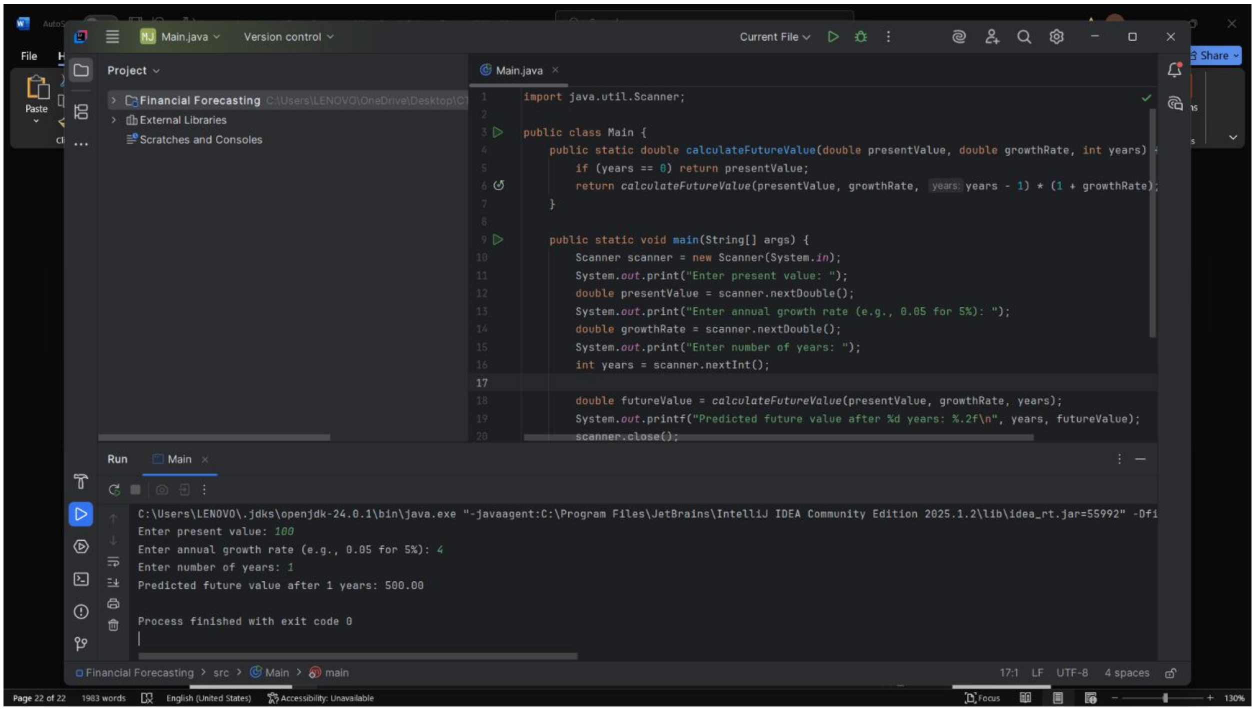
```
C:\Users\LENOVO\.jdks\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.2\lib\idea_rt.jar=55975" -Dfi
Enter title to search:
Algorithms
Choose search method:
1. Linear Search
2. Binary Search
1
Book Found: Book ID: B4, Title: Algorithms, Author: Robert Sedgewick

Process finished with exit code 0
```

Exercise 7: Financial Forecasting

```
import java.util.Scanner;

public class Main {
  public static double calculateFutureValue(double presentValue, double growthRate, int years) {
    if (years == 0) return presentValue;
    return calculateFutureValue(presentValue, growthRate, years - 1) * (1 + growthRate);
  }

  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter present value: ");
    double presentValue = scanner.nextDouble();
    System.out.print("Enter annual growth rate (e.g., 0.05 for 5%): ");
    double growthRate = scanner.nextDouble();
    System.out.print("Enter number of years: ");
    int years = scanner.nextInt();
```

```
        double futureValue = calculateFutureValue(presentValue, growthRate, years);
        System.out.printf("Predicted future value after %d years: %.2f\n", years, futureValue);
        scanner.close();
    }
}
```

Output:



Main.java
Version control
Current File
Project
Financial Forecasting C:\Users\LENOVO\OneDrive\Desktop\C
External Libraries
Scratches and Consoles
Run
Main
Financial Forecasting > src > Main > main
17:1 LF UTF-8 4 spaces
Page 22 of 22 1983 words English (United States) Accessibility: Unavailable
Focus
130%

```
import java.util.Scanner;

public class Main {
    public static double calculateFutureValue(double presentValue, double growthRate, int years)
        if (years == 0) return presentValue;
        return calculateFutureValue(presentValue, growthRate, years: years - 1) * (1 + growthRate);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter present value: ");
        double presentValue = scanner.nextDouble();
        System.out.print("Enter annual growth rate (e.g., 0.05 for 5%): ");
        double growthRate = scanner.nextDouble();
        System.out.print("Enter number of years: ");
        int years = scanner.nextInt();

        double futureValue = calculateFutureValue(presentValue, growthRate, years);
        System.out.printf("Predicted future value after %d years: %.2f\n", years, futureValue);
        scanner.close();
```

```
C:\Users\LENOVO\.jdks\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.2\lib\idea_rt.jar=55992" -Dfi
Enter present value: 100
Enter annual growth rate (e.g., 0.05 for 5%): 4
Enter number of years: 1
Predicted future value after 1 years: 500.00

Process finished with exit code 0
```