**Algorithms Data Structures - Handson Exercises**

**Exercise 1: Inventory Management System**

**Code:**

//Product.java

**package** InventoryManagement;

**public** **class** Product{

String productId;

String productName;

**int** quantity;

**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 toString(){

**return** "ProductID:"+productId+",Name:" +productName+", Qty:"+ quantity+", Price:"+price;

}

}

// InventoryManager.java

**package** InventoryManagement;

**import** java.util.HashMap;

**public** **class** InventoryManager{

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

**public** **void** addProduct(Product product){

inventory.put(product.productId, product);

}

**public** **void** updateProduct(String productId, **int** newQty, **double** newPrice){

**if**(inventory.containsKey(productId)){

Product p = inventory.get(productId);

p.quantity = newQty;

p.price = newPrice;

} **else**{

System.***out***.println("Product not found!");

}

}

**public** **void** deleteProduct(String productId){

inventory.remove(productId);

}

**public** **void** displayAll(){

**for**(Product p : inventory.values()){

System.***out***.println(p);

}

}

**public** **static** **void** main(String[] args){

InventoryManager manager = **new** InventoryManager();

manager.addProduct(**new** Product("1", "Keyboard", 50, 999.9));

manager.addProduct(**new** Product("2", "Mouse", 100, 499.49));

manager.displayAll();

manager.updateProduct("1", 70, 899.00);

manager.deleteProduct("2");

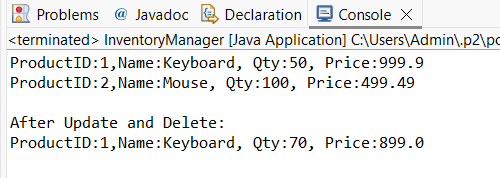
System.***out***.println("\nAfter Update and Delete:");

manager.displayAll();

}

}

**Output:**



**Exercise 3: Sorting Customer Orders**

**Code:**

// Order.java

**package** sortingorders;

**public** **class** Order{

**public** **int** orderId;

**public** String customerName;

**public** **double** totalPrice;

**public** Order(**int** 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: Rs." + totalPrice;

}

}

// BubbleSort.java

**package** sortingorders;

**public** **class** BubbleSort{

**public** **static** **void** sort(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;

}

}

}

}

}

// QuickSort.java

**package** sortingorders;

**public** **class** QuickSort{

**public** **static** **void** sort(Order[] orders, **int** low, **int** high){

**if** (low < high){

**int** pi = *partition*(orders, low, high);

*sort*(orders, low, pi - 1);

*sort*(orders, pi + 1, high);

}

}

**private** **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;

}

}

// Main.java

**package** sortingorders;

**public** **class** Main{

**public** **static** **void** main(String[] args){

Order[] orders = {

**new** Order(101, "Alice", 150.75),

**new** Order(102, "Bob", 99.99),

**new** Order(103, "Charlie", 299.49),

**new** Order(104, "David", 175.20)

};

System.***out***.println("Original Orders:");

**for** (Order order : orders)

System.***out***.println(order);

BubbleSort.*sort*(orders);

System.***out***.println("\nSorted using Bubble Sort:");

**for** (Order order : orders)

System.***out***.println(order);

Order[] quickOrders = {

**new** Order(101, "Alice", 150.75),

**new** Order(102, "Bob", 99.99),

**new** Order(103, "Charlie", 299.49),

**new** Order(104, "David", 175.20)

};

QuickSort.*sort*(quickOrders, 0, quickOrders.length - 1);

System.***out***.println("\nSorted using Quick Sort:");

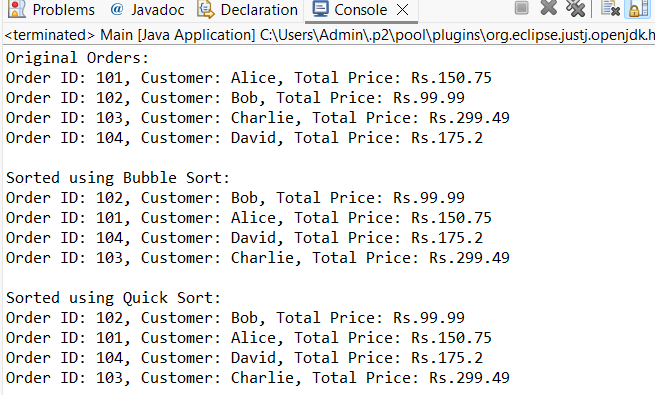
**for** (Order order : quickOrders)

System.***out***.println(order);

}

}

**Output:**



**Exercise 4: Employee Management System**

**Code:**

// Employee.java

**package** management.employee;

**public** **class** Employee{

**public** **int** employeeId;

**public** String name;

**public** String position;

**public** **double** salary;

**public** Employee(**int** 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;

}

}

// EmployeeSystem.java

**package** management.employee;

**import** java.util.Scanner;

**public** **class** EmployeeSystem{

**static** Employee[] *employees* = **new** Employee[100];

**static** **int** *count* = 0;

**public** **static** **void** addEmployee(Employee e){

*employees*[*count*++] = e;

}

**public** **static** **void** searchEmployee(**int** id){

**for** (**int** i = 0; i < *count*; i++){

**if** (*employees*[i].employeeId == id){

System.***out***.println(*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(**int** id){

**for** (**int** i = 0; i < *count*; i++){

**if** (*employees*[i].employeeId == id){

**for** (**int** j = i; j < *count* - 1; j++){

*employees*[j] = *employees*[j + 1];

}

*count*--;

System.***out***.println("Employee deleted.");

**return**;

}

}

System.***out***.println("Employee not found.");

}

**public** **static** **void** main(String[] args){

*addEmployee*(**new** Employee(1, "Alice", "Manager", 50000));

*addEmployee*(**new** Employee(2, "Bob", "Developer", 40000));

*traverseEmployees*();

*searchEmployee*(2);

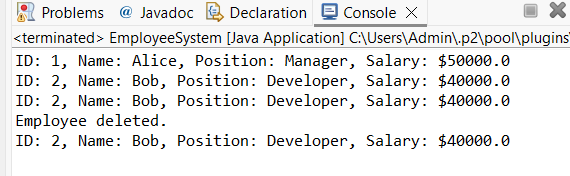
*deleteEmployee*(1);

*traverseEmployees*();

}

}

**Output:**



**Exercise 5: Task Management System**

**Code:**

// Task.java

**package** management.task;

**public** **class** Task{

**public** **int** taskId;

**public** String taskName;

**public** String status;

**public** Task next;

**public** Task(**int** taskId, String taskName, String status){

**this**.taskId = taskId;

**this**.taskName = taskName;

**this**.status = status;

**this**.next = **null**;

}

**public** String toString(){

**return** "Task ID: " + taskId + ", Name: " + taskName + ", Status: " + status;

}

}

// TaskManager.java

**package** management.task;

**public** **class** TaskManager{

Task head = **null**;

**public** **void** addTask(Task task){

**if** (head == **null**){

head = task;

}**else**{

Task current = head;

**while** (current.next != **null**)

current = current.next;

current.next = task;

}

}

**public** **void** traverseTasks() {

Task current = head;

**while** (current != **null**) {

System.***out***.println(current);

current = current.next;

}

}

**public** **void** searchTask(**int** taskId) {

Task current = head;

**while** (current != **null**) {

**if** (current.taskId == taskId) {

System.***out***.println(current);

**return**;

}

current = current.next;

}

System.***out***.println("Task not found.");

}

**public** **void** deleteTask(**int** taskId){

**if** (head == **null**) **return**;

**if** (head.taskId == taskId){

head = head.next;

**return**;

}

Task current = head;

**while** (current.next != **null**){

**if** (current.next.taskId == taskId){

current.next = current.next.next;

**return**;

}

current = current.next;

}

}

**public** **static** **void** main(String[] args){

TaskManager tm = **new** TaskManager();

tm.addTask(**new** Task(1, "Design UI", "Pending"));

tm.addTask(**new** Task(2, "Develop Backend", "In Progress"));

tm.traverseTasks();

tm.searchTask(2);

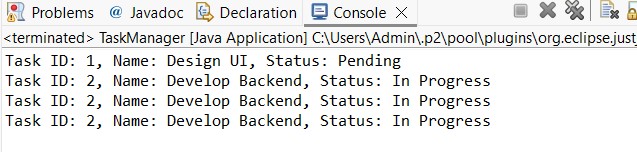
tm.deleteTask(1);

tm.traverseTasks();

}

}

**Output:**



**Exercise 6: Library Management System**

**Code:**

// Book.java

**package** search.library;

**public** **class** Book{

**public** **int** bookId;

**public** String title;

**public** String author;

**public** Book(**int** 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;

}

}

// SearchLibrary.java

**package** search.library;

**import** java.util.Arrays;

**import** java.util.Comparator;

**public** **class** SearchLibrary{

**static** Book[] *books* = {

**new** Book(1, "Data Structures", "Mark Allen"),

**new** Book(2, "Java Basics", "James Gosling"),

**new** Book(3, "Algorithms", "CLRS")

};

**public** **static** **void** linearSearch(String title){

**for** (Book book : *books*){

**if** (book.title.equalsIgnoreCase(title)){

System.***out***.println("Found: " + book);

**return**;

}

}

System.***out***.println("Book not found.");

}

**public** **static** **void** binarySearch(String title){

Arrays.*sort*(*books*, Comparator.*comparing*(b -> b.title));

**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("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){

*linearSearch*("Java Basics");

*binarySearch*("Algorithms");

}

}

**Output:**

