**Exercise 1: Inventory Management System**

**code:**

import java.util.\*;

class Product{

    int productId;

    String productName;

    int quantity;

    double price;

    Product(int x,String y,int a,double b){

        productId=x;

        productName=y;

        quantity=a;

        price=b;

    }

}

class Inventory {

    private Map<Integer,Product> x = new HashMap<>();

    void add(Product p){

        if(x.containsKey(p.productId)){

            System.out.println("The product "+p.productName+" already exists try updating");

        }

        else{

            x.put(p.productId, p);

            System.out.println("Product "+p.productName+" Successfully added!");

        }

    }

    void update(Product p){

        Product q=x.get(p.productId);

        if(q==null){

            System.out.println("The product "+p.productName+" does not exist try adding it");

        }

        else{

            q.productName=p.productName;

            q.quantity=p.quantity;

            q.price=p.price;

        }

    }

    void delete(int id){

        if(x.remove(id)!=null) System.out.println("Product  Removed");

        else System.out.println("The product  does not exist");

    }

    void display(){

        if(x.isEmpty()){

            System.out.println("Inventory Empty");

        }

        else{

            for(Product i : x.values())

                System.out.println(i.productId+" "+i.productName+" "+i.quantity+" "+i.quantity);

        }

    }

}

public class InventoryManagement {

    public static void main(String[] args) {

        Inventory i=new Inventory();

        Product a=new Product(1, "hammer", 10, 20.00);

        Product b=new Product(2, "nails", 20, 1.50);

        Product c=new Product(3, "screws", 30, 2.00);

        Product d=new Product(3, "screws", 40, 2.00);

        i.add(a);

        i.add(b);

        i.add(c);

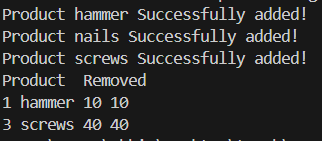
        i.update(d);

        i.delete(2);

        i.display();

    }

}

**output:**

**Exercise 2: E-commerce Platform Search Function**

**code:**

import java.util.\*;

class Product{

    int productId;

    String productName;

    int quantity;

    double price;

    Product(int x,String y,int a,double b){

        productId=x;

        productName=y;

        quantity=a;

        price=b;

    }

}

class linear{

    ArrayList<Product> ar=new ArrayList<>();

    void add(Product p){

        ar.add(p);

    }

    boolean search(int id){

        for(Product i : ar){

            if (i.productId==id) return true;

        }

        return false;

    }

}

class binary{

    ArrayList<Product> ar=new ArrayList<>();

    void add(Product p){

        int lo=0,hi=ar.size()-1;

        while(lo<=hi){

            int mid=(lo+hi)/2;

            if (ar.get(mid).productId<p.productId) lo=mid+1;

            else hi=mid-1;

        }

        ar.add(lo,p);

    }

    boolean search(int id){

        int lo=0,hi=ar.size()-1;

        while(lo<=hi){

            int mid=(lo+hi)/2;

            if (ar.get(mid).productId==id) return true;

            else if (ar.get(mid).productId<id) lo=mid+1;

            else hi=mid-1;

        }

        return false;

    }

}

public class E\_Commerce {

    public static void main(String[] args) {

        Product x=new Product(1, "hammer", 10, 20.00);

        Product y=new Product(2, "nails", 20, 1.50);

        Product z=new Product(3, "screws", 30, 2.00);

        linear l = new linear();

        binary b = new binary();

        l.add(x);

        l.add(y);

        l.add(z);

        b.add(x);

        b.add(y);

        b.add(z);

        if(l.search(2)) System.out.println("The product with id 2 is found using linear search");

        else System.out.println("The product with id 2 is not found using linear search");

        if(b.search(2)) System.out.println("The product with id 2 is found using binary search");

        else System.out.println("The product with id 2 is not found using binary search");

    }

}

**output:**



**Exercise 3: Sorting Customer Orders**

**code:**

import java.util.ArrayList;

class Order{

    int orderId;

    String customerName;

    double totalPrice;

    Order(int x,String y,double z){

        orderId=x;

        customerName=y;

        totalPrice=z;

    }

}

public class SortingCustomerOrders {

    static void bubble(ArrayList<Order> ar){

        int n=ar.size();

        for(int i=0;i<n-1;i++){

            for(int j=0;j<n-i-1;j++){

                if(ar.get(j).orderId>ar.get(j+1).orderId){

                    Order t=ar.get(j);

                    ar.set(j, ar.get(j+1));

                    ar.set(j+1,t);

                }

            }

        }

    }

    static void quick(ArrayList<Order> ar,int lo,int hi){

        if(lo<hi){

            int p=lo;

            int i=lo,j=hi;

            while(i<=j){

                while( i<=hi && ar.get(i).orderId<=ar.get(p).orderId) i++;

                while(ar.get(j).orderId>ar.get(p).orderId) j--;

                if(i<j){

                    Order t=ar.get(j);

                    ar.set(j, ar.get(i));

                    ar.set(i,t);

                }

            }

            Order t=ar.get(j);

            ar.set(j, ar.get(p));

            ar.set(p,t);

            quick(ar,lo,j-1);

            quick(ar,j+1,hi);

        }

    }

    static void display(ArrayList<Order> x){

        for(Order i:x){

            System.out.println("Id : "+i.orderId+" name : "+i.customerName+" price : "+i.totalPrice);

        }

    }

    public static void main(String[] args) {

        ArrayList<Order> x=new ArrayList<>();

        x.add(new Order(5, "abc", 1500.50));

        x.add(new Order(3, "def", 2500.25));

        x.add(new Order(2, "ghi", 2000.75));

        x.add(new Order(1, "jkl", 1000.00));

        x.add(new Order(4, "mno", 500.50));

        ArrayList<Order> y=new ArrayList<>();

        for(Order i : x){

            y.add(new Order(i.orderId, i.customerName, i.totalPrice));

        }

        System.out.println("Sorting using bubble sort ..");

        bubble(x);

        display(x);

        System.out.println("\nSorting using Quick sort ..");

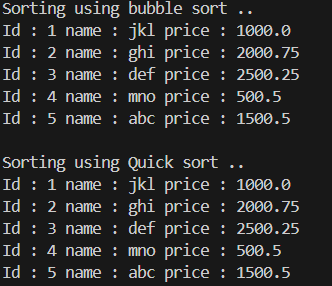
        quick(y, 0, 4);

        display(y);

    }

}

**output:**



**Exercise 4: Employee Management System**

**code:**

class Employee{

    int employeeId;

    String employeeName,position;

    double salary;

    Employee(int id,String name,String pos,double sal){

        employeeId=id;

        employeeName=name;

        position=pos;

        salary=sal;

    }

}

public class EmployeeManagement {

    static Employee[] em = new Employee[5];

    static void add(Employee e){

        for(int i=0;i<5;i++){

            if(em[i]==null) {

                em[i]=e;

                return;

            }

        }

        System.out.println("Memory full!");

    }

    static Employee search(int id){

        for(int i=0;i<5;i++){

            if(em[i]!=null && em[i].employeeId==id) return em[i];

        }

        return null;

    }

    static void traverse(){

        for(int i=0;i<5;i++){

            if(em[i]!=null)

            System.out.println("ID : "+em[i].employeeId+" name : "+em[i].employeeName+" Position : "+em[i].position+" Salary : "+em[i].salary);

        }

    }

    static void delete(int id){

        for(int i=0;i<5;i++){

            if(em[i]!=null && em[i].employeeId==id) {

                System.out.println("Employee "+em[i].employeeName+" deleted");

                em[i]=null;

            }

        }

    }

    public static void main(String[] args) {

        for(int i=0;i<5;i++){

            em[i]=null;

        }

        add(new Employee(1, "abc", "mngr", 50000));

        add(new Employee(2, "def", "asstmngr", 35000));

        add(new Employee(3, "ghi", "asstmngr", 35000));

        add(new Employee(4, "jkl", "tmngr", 25000));

        add(new Employee(5, "mno", "clk", 25000));

        Employee a=search(2);

        if(a!=null) {

            System.out.println("employee details are :");

            System.out.println("ID : "+a.employeeId+" name : "+a.employeeName+" Position : "+a.position+" Salary : "+a.salary);

        }

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

        System.out.println();

        delete(3);

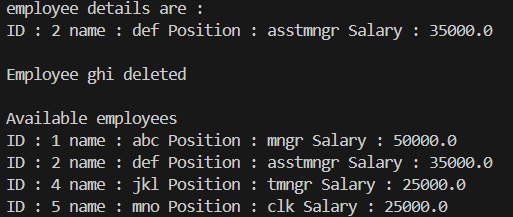
        System.out.println("\nAvailable employees");

        traverse();

    }

}

**output:**



**Exercise 5: Task Management System**

**code:**

class Task{

    int taskId;

    String taskName,status;

    Task nxt;

    Task(int id,String name,String st){

        taskId=id;

        taskName=name;

        status=st;

        nxt=null;

    }

}

public class TaskManagement {

    static Task f=null,l=null;

    static void add(Task t){

        if(f==null && l==null){

            f=t;

            l=t;

        }

        else{

            l.nxt=t;

            l=t;

        }

    }

    static Task search(int id){

        Task x=f;

        while(x!=null){

            if(x.taskId==id) return x;

            x=x.nxt;

        }

        return null;

    }

    static void traverse(){

        Task x=f;

        while(x!=null){

            System.out.println("ID : "+x.taskId+" name : "+x.taskName+" status : "+x.status);

            x=x.nxt;

        }

    }

    static void delete(int id){

        if(f==null) return;

        Task x=f;

        if(x.taskId==id){

            if(x==l){

                f=null;

                l=null;

            }

            else{

                f=f.nxt;

            }

        }

        Task y=x.nxt;

        while(y!=null){

            if(y.taskId==id){

                x.nxt=y.nxt;

                if(y==l){

                    l=x;

                }

                System.out.println("Task "+y.taskName+"is deleted");

            }

            y=y.nxt;

        }

    }

    public static void main(String[] args) {

        add(new Task(1, "sysmgr", "running"));

        add(new Task(5, "awcc", "suspended"));

        add(new Task(2, "chrome", "running"));

        add(new Task(9, "teams", "stopped"));

        Task a=search(9);

        System.out.println("Searching task with id 9");

        if(a!=null){

            System.out.println("Task found ");

            System.out.println("ID : "+a.taskId+" name : "+a.taskName+" status : "+a.status);

        }

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

        System.out.println();

        delete(5);

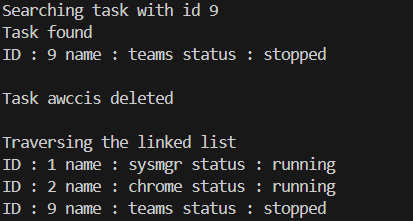
        System.out.println("\nTraversing the linked list");

        traverse();

    }

}

**output:**



**Exercise 6: Library Management System**

**code:**

import java.util.ArrayList;

class Book{

    int bookId;

    String title,author;

    Book(int id,String t,String a){

        bookId=id;

        title=t;

        author=a;

    }

}

public class LibraryManagement {

    static ArrayList<Book> ar1=new ArrayList<>();

    static ArrayList<Book> ar2=new ArrayList<>();

    static void ladd(Book b){

        ar1.add(b);

    }

    static void badd(Book b){

        int lo=0,hi=ar2.size()-1;

        while(lo<=hi){

            int mid=(lo+hi)/2;

            if ((ar2.get(mid).author).compareTo(b.author)<0) lo=mid+1;

            else hi=mid-1;

        }

        ar2.add(lo,b);

    }

    static Book lsearch(String aut){

        for(Book i:ar1){

            if(i.author.equals(aut)){

                return i;

            }

        }

        return null;

    }

    static Book bsearch(String aut){

        int lo=0,hi=ar2.size()-1;

        while(lo<=hi){

            int mid=(lo+hi)/2;

            if (ar2.get(mid).author.equals(aut)) return ar2.get(mid);

            else if (ar2.get(mid).author.compareTo(aut)<0) lo=mid+1;

            else hi=mid-1;

        }

        return null;

    }

    public static void main(String[] args) {

        ladd(new Book(1, "HP goblet of fire", "JK Rowling"));

        ladd(new Book(2, "Eragon", "C paoline"));

        ladd(new Book(3, "Digital Fortress", "Dan brown"));

        ladd(new Book(5, "Sword of Summer", "R Riordan"));

        badd(new Book(1, "HP goblet of fire", "JK Rowling"));

        badd(new Book(2, "Eragon", "C paoline"));

        badd(new Book(3, "Digital Fortress", "Dan brown"));

        badd(new Book(5, "Sword of Summer", "R Riordan"));

        System.out.println("Searching for book with author JK Rowling using linear search");

        Book k=lsearch("JK Rowling");

        if(k!=null){

            System.out.println("Book found ");

            System.out.println("ID : "+k.bookId+" name : "+k.title+" author : "+k.author);

        }

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

        System.out.println("\nSearching for book with author R Riordan using binary search");

        Book l=bsearch("R Riordan");

        if(l!=null){

            System.out.println("Book found ");

            System.out.println("ID : "+l.bookId+" name : "+l.title+" author : "+l.author);

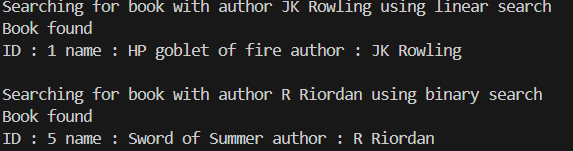
        }

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

    }

}

**output:**



**Exercise 7: Financial Forecasting**

**code:**

public class FinancialForecasting {

    static double predict(double cur,double rate,int years){

        if(years==0) return cur;

        else return(predict(cur\*(1+rate),rate,years-1));

    }

    public static void main(String[] args) {

        System.out.println("Assuming a capital of 50,000 with a rate of increasing of 12%, and with the time period of 5 years :");

        System.out.println(predict(50000, 0.12, 5));

    }

}

**output:**

