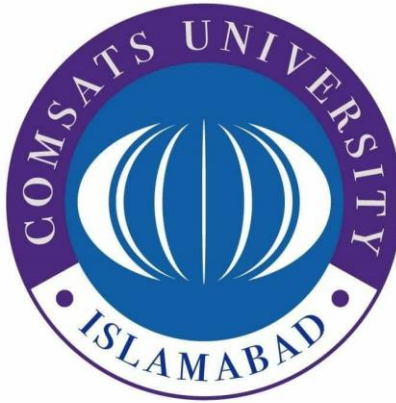


# PHARMACY MANAGEMENT SYSTEM



**SUBMITTED BY:**

**SUDAIS KHAN**

**FA20-BSE-042**

**&**

**ZUNAIRA FATIMA**

**FA20-BSE-014**

**SUBJECT: DATA STRUCTURE & ALGORITHM LAB**

**SECTION: BSE-3B**

**SUBMITTED TO SIR TAHIR**

# STOCKS

**Description:** In our project pharmacy management system I make a **Stock entity** in a **double Linked list**. Where user can **Insert** the stocks in the pharmacy and **Show** the Stocks and **Delete** the stocks and **Search** to the stocks and also give the functionality to insert a stock at **any position**.

```
//doubly linked list
import static java.lang.System.exit;
import java.util.Scanner;
class Node1{
    String data;
    Node1 next;
    Node1 pre;
}
public class Stocks {
    public Node1 insert(Node1 node,String data){
        if(node==null){
            return getnewnode(data);
        }
        Node1 head=node;
        while(node.next!=null){
            node=node.next; }
        Node1 n=getnewnode(data);
        n.pre=node;
        node.next=n;
        return head;}
    public Node1 insertatposition(Node1 node,String data,int pos){
        if(node==null){
            if(pos==1){
                return getnewnode(data);
            }else{
                return null;}
        }
        if(pos==1){
            Node1 t=getnewnode(data);
            t.next=node;
            node.pre=t;
            return t;
        }
        Node1 head=node;
        while(node!=null && pos>2){
            node=node.next;
            pos--;
        }
        if(node==null){
            System.out.println("Invalid position");
            return head;
        }
        Node1 b=getnewnode(data);
```

```

        b.next=node.next;
        b.pre=node;
        if (node.next!=null) {
            node.next.pre=b;
        }
        node.next=b;
        return head;
    }
    public Node1 deleteanyposition(Node1 root,int pos){
        if (root==null) {
            return root;
        }
        if (pos==1) {
            if (root.next!=null) {
                root.next.pre=null;
            }
            return root.next;
        }
        Node1 head=root;
        while (head!=null && pos>1) {
            head=head.next;
            pos--;
        }
        if (head==null) {
            return null;
        }
        if (head.next!=null) {
            head.next.pre=head.pre;
        }
        head.pre.next=head.next;
        return root;
    }
    public Node1 getnewnode(String data){
        Node1 m=new Node1();
        m.data=data;
        m.next=null;
        m.pre=null;
        return m;
    }
    public void print(Node1 node){
        if (node==null) {
            return;
        }
        System.out.println("Stocks of Pharmacy:");
        while (node!=null) {
            System.out.println(node.data);
            node=node.next;
        }
        System.out.println("");
    }
    public void searchNode(String value,Node1 root) {
        int i = 1;
        boolean flag = false;
        //Node current will point to head
        Node1 current =root;
        if (root == null) {
            System.out.println("List is empty");
            return;
        }
    }

```

```

    }
    while(current != null) {
        if(current.data == value) {
            flag = true;
            break; }
        current = current.next;
        i++;}
    if(flag)
System.out.println("This Stock is present in the pharmacy at the position : "
+ i+", Stock Name:"+value);
    else
        System.out.println("This Stock is not present in the pharmacy");
}
public static void main(String[] args) {
    Stocks s=new Stocks();
    Node1 root=null;
    // Inserted Stocks in the pharmacy
    root=s.insert(root, "GETZ");
    root=s.insert(root, "gsk");
    root=s.insert(root, "CVS Health");
    root=s.insert(root, "Pfizer");
    root=s.insert(root, "AbbVie");
    root=s.insert(root, "Walgreens");
    System.out.println("*****MAIN MENU*****");
    System.out.println("Press 1 to Show the Stocks in the Pharmacy");
    System.out.println("Press 2 to Delete the Stocks at any position in the
pharmacy");
    System.out.println("Press 3 to Search the Stocks in the Pharmacy");
    System.out.println("Press 4 to insert the Stocks at position in the
Pharmacy");
    System.out.println("Press 5 to exit");
    Scanner sc=new Scanner(System.in);
    while(sc.hasNext()){
        int num=sc.nextInt();
        switch(num){
            case 1:
                s.print(root);
                break;
            case 2:
                root=s.deleteanyposition(root, 1);
                s.print(root);
                break;
            case 3:
                s.searchNode("Pfizer", root);
                break;
            case 4:
                root=s.insertatposition(root, "Amerisource", 1);
                s.print(root);
                break;
            case 5:
                exit(0);
            default:
                System.out.println("Invalid Input"); }}}}

```

# MEDICINE

**DESCRIPTION:** In Our Project Pharmacy Management System I made a **Medicine Entity** in a **Singly Linked List** here the user can easily **Insert** the medicines in the pharmacy and it also give the functionality to **Delete** the Medicine from the pharmacy.

```
import static java.lang.System.exit;
import java.util.Scanner;
class Node{
    String data;
    Node next;
}
public class medicine {
    Node head;
    // insertion
    public void insert(String data){
        Node node=new Node();
        node.data=data;
        node.next=null;
        if(head==null){
            head=node;
        }
        else{
            Node n=head;
            while(n.next!=null){
                n=n.next;
            }
            n.next=node;
        }
    }
    // Deletion
    public void deleteAt(int index){
        if(index==0){
            head=head.next;
        }else{
            Node n=head;
            Node n1=null;
            for(int i=0;i<index-1;i++){
                n=n.next;
            }
            n1=n.next;
            n.next=n1.next;
            System.out.println("Deleted This Medicine From Pharmacy---
>" +n1.data);
        }
    }
    public void show(){
        Node node=head;
        while(node.next!=null){
            System.out.println(node.data);
        }
    }
}
```

```

        node=node.next;
    }
    System.out.println(node.data);
}

public static void main(String[] args) {
    medicine m = new medicine();
    // Insertion here
    m.insert("Panadol 500mg");
    m.insert("Extor 80mg");
    m.insert("Risek 20mg");
    m.insert("Sitaglu Met 50mg");
    m.insert("Lipirex 10mg");
    m.insert("Getryl 2mg");
    m.insert("Loprin 75m");

    Scanner sc=new Scanner(System.in);
    System.out.println("*****MAIN MENU*****");
    System.out.println("Press 1 to Show Medicine In the Pharmacy");
    System.out.println("Press 2 to Delete Medicine from the Pharmacy");
    System.out.println("Press 3 to exit");
    while(sc.hasNext()){
        int num=sc.nextInt();

        switch(num){
            case 1:
                m.show();
                break;
            case 2:
                m.deleteAt(6);
                m.show();
                System.out.println("");
                break;
            case 3:
                exit(0);
            default:
                System.out.println("Invalid User Input");
        }
    }
}
}

```

## Regular Patient

**Description:** In our project Pharmacy Management System I make a **Regular Patient** Entity in a **Queue** Data Structure where I give the functionality that the **first patient come in the pharmacy will be treated first** and also user can **show** the patient name and also user can **pop** them from the pharmacy.

```
import static java.lang.System.exit;
import java.util.Scanner;
public class RegularPatient {
    String queue[]=new String[100];
    int size;
    int front; int rear;
    public void enqueue(String data){
        queue[rear]=data;
        rear++;
        size=size+1; }
    public void dequeue(){
        String data=queue[front];
        front++;
        size--;
        System.out.println(data+" is pop from Pharmacy");}
    public void display(){
        for(int i=0;i<size;i++){
            System.out.println(queue[i]+""); } }
    public static void main(String[] args) {
        RegularPatient rp=new RegularPatient();
        rp.enqueue("WAQAR");
        rp.enqueue("ALI");
        rp.enqueue("ASIF");
        rp.enqueue("NEELUM");
        rp.enqueue("MEHWISH");
        rp.enqueue("ZAINUB");
        Scanner sc=new Scanner(System.in);
        System.out.println("*****MAIN MENU*****");
        System.out.println("Press 1 to show the Patient");
        System.out.println("Press 2 to pop the patient");
        System.out.println("Press 3 to exit");
        int num=sc.nextInt();
        switch (num) {
            case 1:
                System.out.println("List of Patients:");
                rp.display();
                break;
            case 2:
                rp.dequeue();
                break;
            case 3:
                exit(0);
            default:
                System.out.println("Invalid user input");}}}
```

## Disable Patient

**Description:** In our project Pharmacy management system I make a **Disable Patient** entity in a **Stack** Data Structure where I give the functionality that the **Disable patient come in the pharmacy will be treated first and after the remaining regular patient will be treated**. And the user can also **show** the names of Disable patient and can **pop** them from pharmacy.

```
import static java.lang.System.exit;
import java.util.Scanner;
public class DisablePatient {
    int top=0;
    int max=10;
    String arr[]=new String[max];
    public void insert(String val){
        if(isFull()){
            System.out.println("Pharmacy is Full for disabled seats");
        }
        arr[top++]=val;
    }
    public boolean isFull(){
        if(top==max){
            return true;
        }
        return false;
    }
    public String pop(){
        if(isEmpty()){
            System.out.println("Pharmacy seats is Empty for disabled person");
        }
        return arr[--top];
    }
    public boolean isEmpty(){
        if(top==0){
            return true;
        }
        return false;
    }
    public void show(){
        for(int i=0;i<top;i++){
            System.out.println(arr[i]);
        }
    }
    public static void main(String[] args) {
        DisablePatient c=new DisablePatient();

        c.insert("Ali");
    }
}
```



```

        c.insert("Hassan");
        c.insert("Ayesha");
        c.insert("Alia");
    // Here disbled person will manage first and then the others
    Scanner sc=new Scanner(System.in);
    System.out.println("*****MAIN MENU*****");
    System.out.println("Press 1 to show the names of disabled Person:");
    System.out.println("Press 2 to Pop the disabled person from the
pharmacy:");
    System.out.println("Press 3 to exit");
    while(sc.hasNext()){
        int num=sc.nextInt();
        switch(num){
            case 1:
                System.out.println("List of Disabled Person:");
                c.show();
                break;
            case 2:
                System.out.println( c.pop()+" is pop from the pharmacy ");
                c.show();
                break;
            case 3:
                exit(0);
            default:
                System.out.println("Invalid user input");
        }
    }
}
}
}

```

# Pharmacy

**Description:** In our project Pharmacy management system I make a pharmacy Entity In a **Circular Linked List** Data Structure and give **the functionality just to interlinked the Branches of pharmacy to each other**. And the user can **show** the names of pharmacy that are linked with each other and also user can **Search the pharmacy name at a Position**.

```
import static java.lang.System.exit;
import java.util.Scanner;
public class pharmacy {
    public class Node{
        String data;
        Node next;
        public Node(String data) {
            this.data = data; }
    }
    public Node head = null;
    public Node tail = null;
    //This function will add the new node at the end of the list.
    public void add(String data){
        Node newNode = new Node(data);
        if(head == null) {
            //If list is empty, both head and tail would point to new node.
            head = newNode;
            tail = newNode;
            newNode.next = head;
        }
        else {
            //tail will point to new node.
            tail.next = newNode;
            //New node will become new tail.
            tail = newNode;
            //Since, it is circular linked list tail will point to head.
            tail.next = head; }
    }
    public void search(String element) {
        Node current = head;
        int i = 1;
        boolean flag = false;
        //Checks whether list is empty
        if(head == null) {
            System.out.println("List is empty");
        }
        else {
            do{
                if(current.data==element) {
                    flag = true;
                    break; }
            }
        }
    }
}
```

```

        current = current.next;
        i++;
    }while(current != head);
    if(flag)
        System.out.println("This pharmacy is present at the position : "+ i + "
Pharmacy Name: "+current.data);
    else
        System.out.println("This Pharmacy is not present in the list that are
interlinked with eachother");
    }
}
//Displays all the nodes in the list
public void display() {
    Node current = head;
    if(head == null) {
        System.out.println("List is empty");
    }
    else {
        System.out.println("Pharmacies that are interlinked with each
other are: ");
        do{
            System.out.print(" "+ current.data+"----->");
            current = current.next;
        }while(current != head);
        System.out.println();
    }
}
public static void main(String[] args) {
    pharmacy p = new pharmacy();
    p.add("ALSHIFA");
    p.add("CareFirst");
    p.add("Medlife");
    p.add("MedSavvy");
    p.add("OptumRx");
    p.add("The Pill Club");
    Scanner sc=new Scanner(System.in);
    System.out.println("*****MAIN MENU*****");
    System.out.println("Press 1 to Show the pharmacy that are interlinked
with eachother");
    System.out.println("Press 2 to Search the pharmacy position");
    System.out.println("Press 3 to exit");
    while(sc.hasNext()){
        int num=sc.nextInt();
        switch(num){
            case 1:
                p.display();
                break;
            case 2:
                p.search("Medlife");
                break;
            case 3:
                exit(0);
                break;
            default:
                System.out.println("Invalid user Input");} } }

```