



9/28/2024

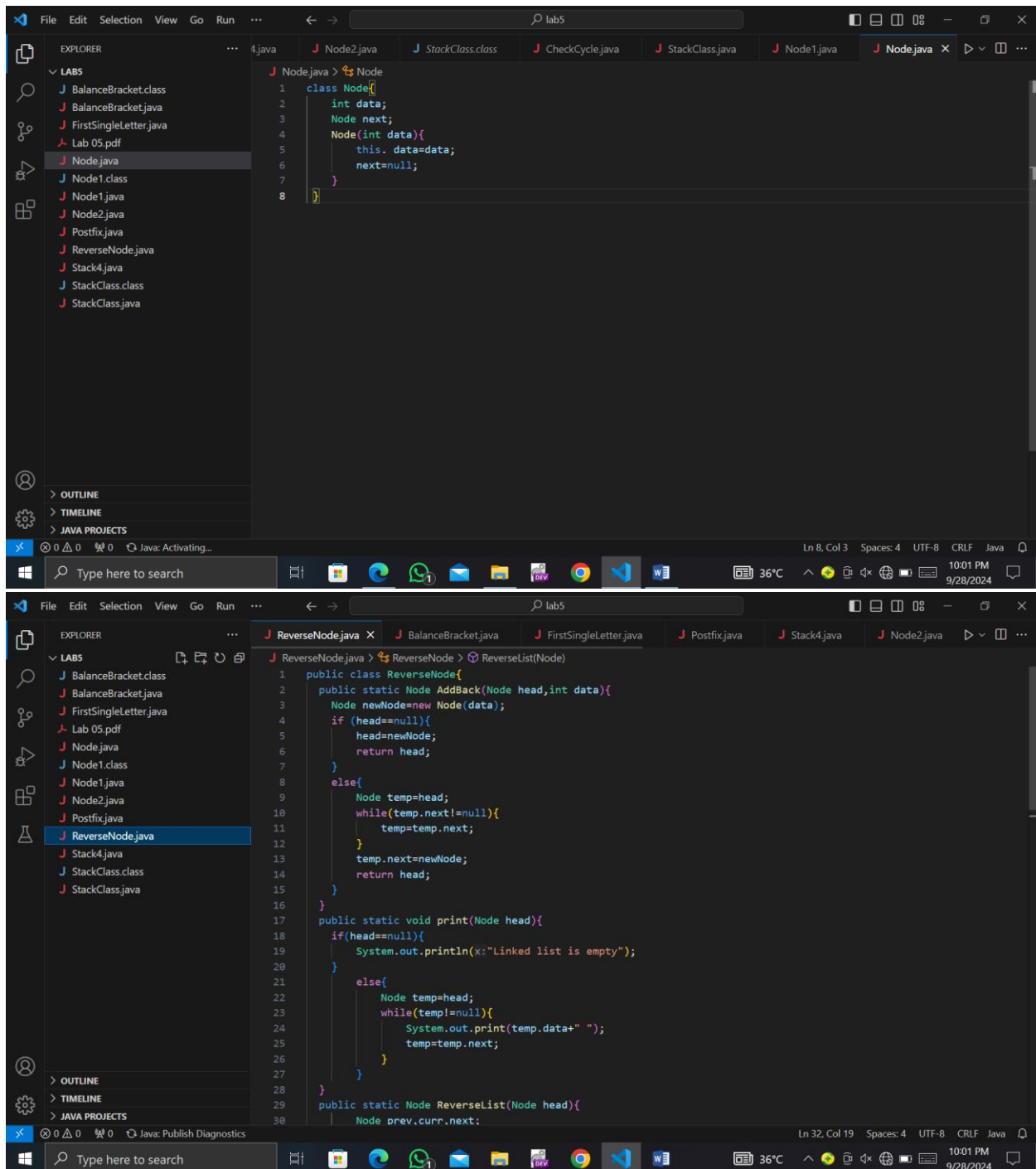
Data Structures

Lab 5 Exercise Solutions

Uzair Ali Memon S/O Yar Muhammad Memon
B-S CS III Section A
023-23-0350
To: Ma'am Marina Rajpoot

Task #1

Solution:



The image displays two screenshots of a Visual Studio Code editor interface, showing Java code for a linked list implementation.

Top Screenshot: The Explorer pane on the left shows a project named "LABS" containing files like BalanceBracket.class, BalanceBracket.java, FirstSingleLetter.java, Lab 05.pdf, Node.java, Node1.class, Node1.java, Node2.java, Postfix.java, ReverseNode.java, Stack4.java, StackClass.class, and StackClass.java. The main editor area displays the code for `Node.java`, which defines a `Node` class with an `int data` field and a `Node next` field. The constructor `Node(int data)` initializes `this.data = data` and `next = null`.

```
1 class Node{
2     int data;
3     Node next;
4     Node(int data){
5         this.data=data;
6         next=null;
7     }
8 }
```

Bottom Screenshot: The Explorer pane shows the same project, but the main editor area displays the code for `ReverseNode.java`. This file contains a `ReverseNode` class with methods for adding a new node to the end of the list (`AddBack`), printing the list (`print`), and reversing the list (`ReverseList`).

```
1 public class ReverseNode{
2     public static Node AddBack(Node head,int data){
3         Node newNode=new Node(data);
4         if (head==null){
5             head=newNode;
6             return head;
7         }
8         else{
9             Node temp=head;
10            while(temp.next!=null){
11                temp=temp.next;
12            }
13            temp.next=newNode;
14            return head;
15        }
16    }
17    public static void print(Node head){
18        if(head==null){
19            System.out.println("Linked list is empty");
20        }
21        else{
22            Node temp=head;
23            while(temp!=null){
24                System.out.print(temp.data+" ");
25                temp=temp.next;
26            }
27        }
28    }
29    public static Node ReverseList(Node head){
30        Node prev,curr,next;
```

The screenshot shows an IDE with the Explorer pane on the left displaying a project named 'LABS'. The file 'ReverseNode.java' is selected. The main editor displays the code for 'ReverseNode' and 'ReverseList(Node)'. The code implements a linked list reversal algorithm using a while loop to traverse the list and reverse the pointers. The 'main' method demonstrates the usage of the 'ReverseList' method.

```
31 curr=head;
32 prev=null;
33 while(curr!=null){
34     next=curr.next;
35     curr.next=prev;
36     prev=curr;
37     curr=next;
38 }
39 return prev;
40 }
41 public static void printReverse(Node head){
42     if(head==null){
43         System.out.println("Linked list is empty");
44     }
45     else{
46         Node temp=head;
47         while(temp!=null){
48             System.out.print(temp.data+" ");
49             temp=temp.next;
50         }
51     }
52 }
53 Run | Debug
54 public static void main(String[] args) {
55     Node head=null;
56     head=AddBack(head,data:1);
57     head=AddBack(head,data:2);
58     head=AddBack(head,data:3);
59     head=AddBack(head,data:4);
60     head=AddBack(head,data:5);
61 }
```

The screenshot shows the same IDE with the 'ReverseNode.java' file. The 'TERMINAL' pane at the bottom displays the output of the program. The program successfully reverses the linked list, printing the data in reverse order: '5 4 3 2 1'.

```
60 print(head);
61 System.out.println();
62 Node head1=ReverseList(head);
63 printReverse(head1);
64 }
65 }
66 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS Run: ReverseNode + - - - - -

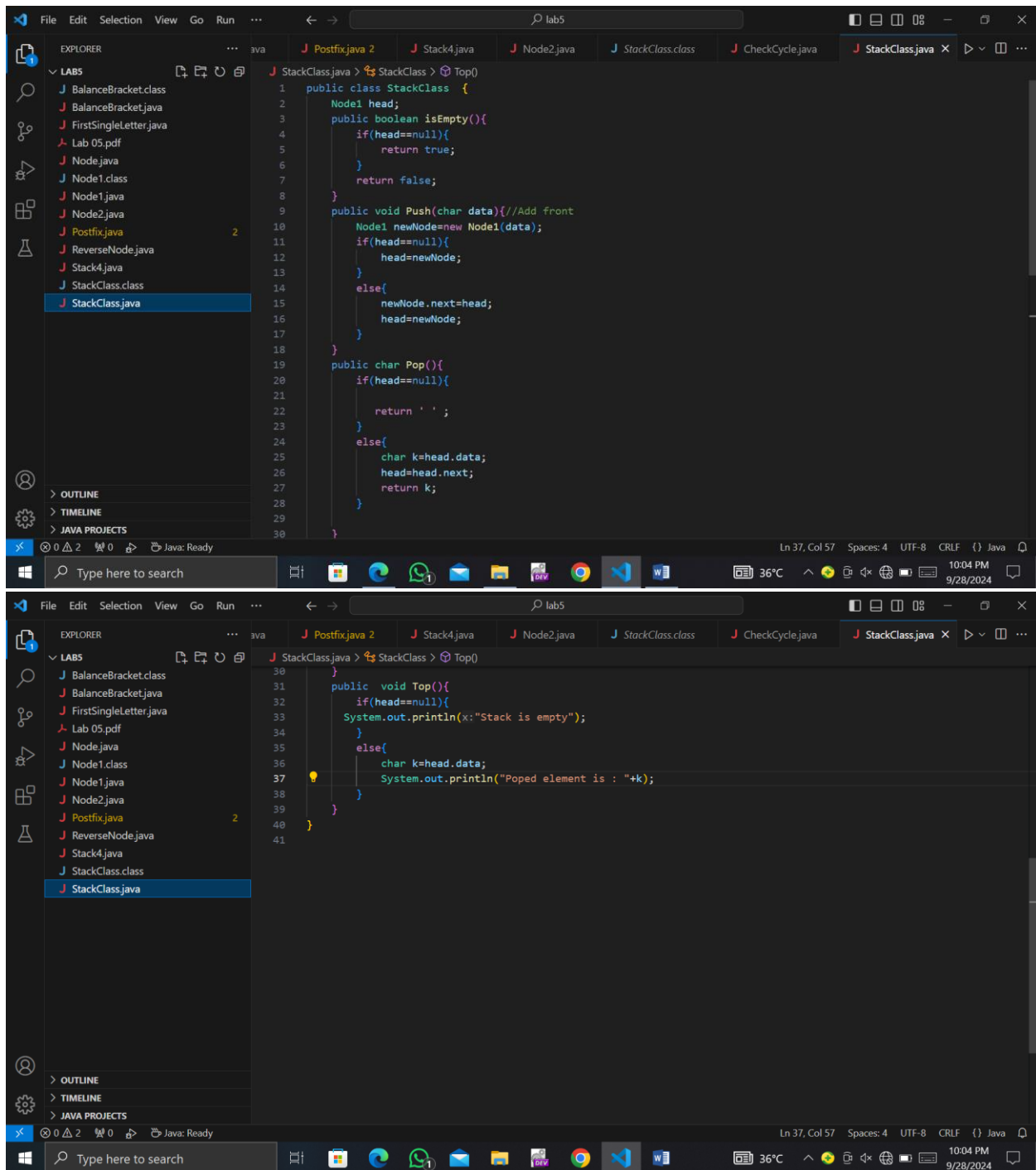
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS E:\bscs3\DSA lab\lab5> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
'C:\Users\8055\AppData\Roaming\Code\User\workspaceStorage\7184f83fe27876845579f80ceef6c5baf\redhat.java\jdt_ws\lab5_e95389
2f\bin' 'ReverseNode'
1 2 3 4 5
5 4 3 2 1
PS E:\bscs3\DSA lab\lab5>
```

Task #2

Solution:



The image displays two screenshots of a Visual Studio Code editor window, showing the implementation of a Stack class in Java. The editor is open to the file `StackClass.java` in the `StackClass` package.

First Screenshot (Top): Shows the `StackClass` implementation with the following methods:

```
1 public class StackClass {
2     Node1 head;
3     public boolean isEmpty(){
4         if(head==null){
5             return true;
6         }
7         return false;
8     }
9     public void Push(char data){//Add front
10        Node1 newNode=new Node1(data);
11        if(head==null){
12            head=newNode;
13        }
14        else{
15            newNode.next=head;
16            head=newNode;
17        }
18    }
19    public char Pop(){
20        if(head==null){
21            return ' ';
22        }
23        else{
24            char k=head.data;
25            head=head.next;
26            return k;
27        }
28    }
29 }
30
```

Second Screenshot (Bottom): Shows the `StackClass` implementation with an additional `Top()` method:

```
30 }
31 public void Top(){
32     if(head==null){
33         System.out.println("Stack is empty");
34     }
35     else{
36         char k=head.data;
37         System.out.println("Popped element is : "+k);
38     }
39 }
40 }
41
```

The screenshot shows an IDE with a file explorer on the left containing a project named 'LABS'. The main editor displays the 'BalanceBracket.java' file. The code defines a 'BalanceBracket' class with a 'Balanced(String str)' method. This method iterates through the string, pushing opening brackets onto a 'StackClass' and popping them when closing brackets are encountered. If the stack is not empty at the end, it prints 'Brackets are un balanced'; otherwise, it prints 'Brackets are balanced'.

```
1 public class BalanceBracket{
2     public void Balanced(String str){
3         StackClass stack=new StackClass();
4         for(int i=0;i<str.length();i++){
5             if(str.charAt(i)=='('||str.charAt(i)=='['||str.charAt(i)=='{'){
6                 stack.Push(str.charAt(i));
7             }
8         }
9         for(int i=0;i<str.length();i++){
10            if(str.charAt(i)==')'){
11                if(stack.Pop()!='('){
12                    System.out.println(x:"Brackets are un balanced");
13                    return;
14                }
15            }
16            else if(str.charAt(i)==']'){
17                if(stack.Pop()!='['){
18                    System.out.println(x:"Brackets are un balanced");
19                    return;
20                }
21            }
22            else if(str.charAt(i)=='}'){
23                if(stack.Pop()!='{'){
24                    System.out.println(x:"Brackets are un balanced");
25                    return;
26                }
27            }
28        }
29        if(!stack.isEmpty()){
30            System.out.println(x:"Brackets are un balanced");
31        }
32        System.out.println(x:"Brackets are balanced");
33    }
34 }
35
36 public static void main(String[] args) {
37     String str="([{}])";
38     BalanceBracket b= new BalanceBracket();
39     b.Balanced(str);
40 }
41
42 }
43 }
```

This screenshot shows the same IDE after running the program. The 'main' method is now visible in the editor. The 'TERMINAL' pane at the bottom shows the command prompt output, which includes the message 'Brackets are un balanced'.

```
31     System.out.println(x:"Brackets are un balanced");
32     return;
33 }
34 System.out.println(x:"Brackets are balanced");
35
36 }
37
38 Run | Debug
39 public static void main(String[] args) {
40     String str="([{}])";
41     BalanceBracket b= new BalanceBracket();
42     b.Balanced(str);
43 }
44 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS E:\bscs3\DSA lab\lab5> & "C:\Program Files\Java\jdk-21\bin\java.exe" "-XX:+ShowCodeDetailsInExceptionMessages" "-cp" "C:\Users\BOSS\AppData\Roaming\Code\User\workspaceStorage\7184f03fe27876845579f0ceef6c5baF\redhat.java\jdt_ws\lab5_e953892f\bin" "BalanceBracket"
Brackets are un balanced
PS E:\bscs3\DSA lab\lab5>
```

Task #3

Solution:

The image shows a screenshot of an IDE (Visual Studio Code) with a Java project named 'lab5'. The Explorer panel on the left shows the project structure, including files like 'BalanceBracket.class', 'BalanceBracket.java', 'FirstSingleLetter.java', 'Lab 05.pdf', 'Node.java', 'Node1.class', 'Node1.java', 'Node2.java', 'Postfix.java', 'ReverseNode.java', 'Stack4.java', 'StackClass.class', and 'StackClass.java'. The main editor displays the code for 'FirstSingleLetter.java', which implements a method to find the first single character in a string. The code is as follows:

```
1 public class FirstSingleLetter {
2     public static void count(String str){
3         char ch=' ';
4         int arr[]=new int[26];
5         for(int j=0;j<arr.length;j++){
6             arr[j]=0;
7         }
8         for(int i=0;i<str.length();i++){
9             ch=str.charAt(i);
10            arr[(int) ch-(int)('A')]+=1;
11        }
12        for(int k=0;k<str.length();k++){
13            ch=str.charAt(k);
14            if(arr[(int)ch-(int)('A')]==1){
15                System.out.println("The First Single character is : "+ch);
16                break;
17            }
18            if(k==str.length()-1){
19                System.out.println("No any element occurs once in string");
20            }
21        }
22    }
23    Run | Debug
24    public static void main(String args[]){
25        String str="ADOADOB";
26        count(str);
27    }
28 }
```

The bottom panel shows the output of the program, which is:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS E:\bscs3\DSA lab\lab5> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\BOSS\AppData\Roaming\Code\User\workspaceStorage\7104f03fe27876845579f0ceef6c5baf\redhat.java\jdt_ws\lab5_e953892f\bin' 'FirstSingleLetter'
The First Single character is : B
PS E:\bscs3\DSA lab\lab5>
```

Task #4

Solution:

The image displays two screenshots of the Visual Studio Code editor interface, showing the implementation of a linked list and a stack.

Top Screenshot: The Explorer panel on the left shows a project named "LABS" with files including `BalanceBracket.class`, `BalanceBracket.java`, `FirstSingleLetter.java`, `Lab 05.pdf`, `Node.java`, `Node1.class`, `Node1.java`, `Node2.java` (selected), `Postfix.java`, `ReverseNode.java`, `Stack4.java`, `StackClass.class`, and `StackClass.java`. The main editor displays the code for `Node2.java`:

```
1 public class Node2 {
2     char data;
3     Node2 next;
4     Node2(char data){
5         this.data=data;
6         next=null;
7     }
8 }
9
```

Bottom Screenshot: The Explorer panel shows the same project with `Stack4.java` selected. The main editor displays the code for `Stack4.java`:

```
1 Stack4.java > Stack4 > push(char)
2 public class Stack4 {
3     Node2 top;
4
5     char peek(){
6         if(top != null)
7         {
8             return top.data;
9         }
10        else
11            return ' ';
12    }
13    void push(char data){
14        Node2 newNode = new Node2(data);
15        newNode.next = top;
16        top = newNode;
17    }
18
19    char pop(){
20        if(top != null){
21            char TopData = top.data;
22            top = top.next;
23            return TopData;
24        }
25        else{
26            return ' ';
27        }
28    }
29 }
30
```


The screenshot shows an IDE window with the file `Postfix.java` open. The Explorer pane on the left lists files under a 'LABS' folder, including `BalanceBracket.class`, `BalanceBracket.java`, `FirstSingleLetter.java`, `Lab 05.pdf`, `Node.java`, `Node1.class`, `Node1.java`, `Node2.java`, `Postfix.java` (selected), `ReverseNode.java`, `Stack4.java`, `StackClass.class`, and `StackClass.java`. The main editor displays the following Java code:

```
1 import java.util.Stack;
2 public class Postfix {
3     public static int precedence(char ch){
4         if(ch=='+'||ch=='-')
5             return 1;
6         else if (ch=='/'||ch=='*')
7             return 2;
8         return -1;
9     }
10    public static void Postfix(String str){
11        Stack4 stack=new Stack4();
12        String result=" ";
13        for(int i=0;i<str.length();i++){
14            if(str.charAt(i)>='a'&& str.charAt(i)<='z' || str.charAt(i)>='A'&& str.charAt(i)<='Z' || str.charAt(i)>=0&& str.charAt(i)<=9){
15                result=result+str.charAt(i);
16            }
17            else if((str.charAt(i) == '+' || (str.charAt(i) == '-' || (str.charAt(i) == '/' || (str.charAt(i) == '*' || (str.charAt(i) == '(' || (str.charAt(i) == ')'))))){
18                if(stack.top == null){
19                    stack.push(str.charAt(i));
20                }
21                else{
22                    while(precedence(stack.peek()) >= precedence(str.charAt(i))){
23                        result = result + stack.pop();
24                    }
25                    stack.push(str.charAt(i));
26                }
27            }
28        }
29        result = result + stack.pop();
30    }
}
```

The status bar at the bottom indicates 'Ln 37, Col 30', 'Spaces: 4', 'UTF-8', 'CRLF', and 'Java'.

The screenshot shows the same IDE window, but now the `main` method is visible, and the terminal at the bottom shows the execution output.

```
30    }
31    while(stack.top!=null){
32        result=result+stack.pop();
33    }
34    System.out.println(result);
35 }
36 public static void main(String[] args) {
37     String str="A+*/BC+C-";
38     Postfix(str);
39 }
40 }
41 }
```

The terminal output shows the execution of the program:

```
PS E:\bscs3\DSA lab\lab5> & "C:\Program Files\Java\jdk-21\bin\java.exe" "-XX:+ShowCodeDetailsInExceptionMessages" "-cp" "C:\Users\rs\BOSS\AppData\Roaming\Code\User\workspaceStorage\7184f03fe27876845579f0ceef6c5baF\redhat.java\jdt_ws\lab5_e953892f\bin" "Postfix"
A*BC/+C+-
PS E:\bscs3\DSA lab\lab5>
```

The status bar at the bottom indicates 'Ln 35, Col 6', 'Spaces: 4', 'UTF-8', 'CRLF', and 'Java'.

THE END