Thaneesha Fender

B1 batch

Roll no.7

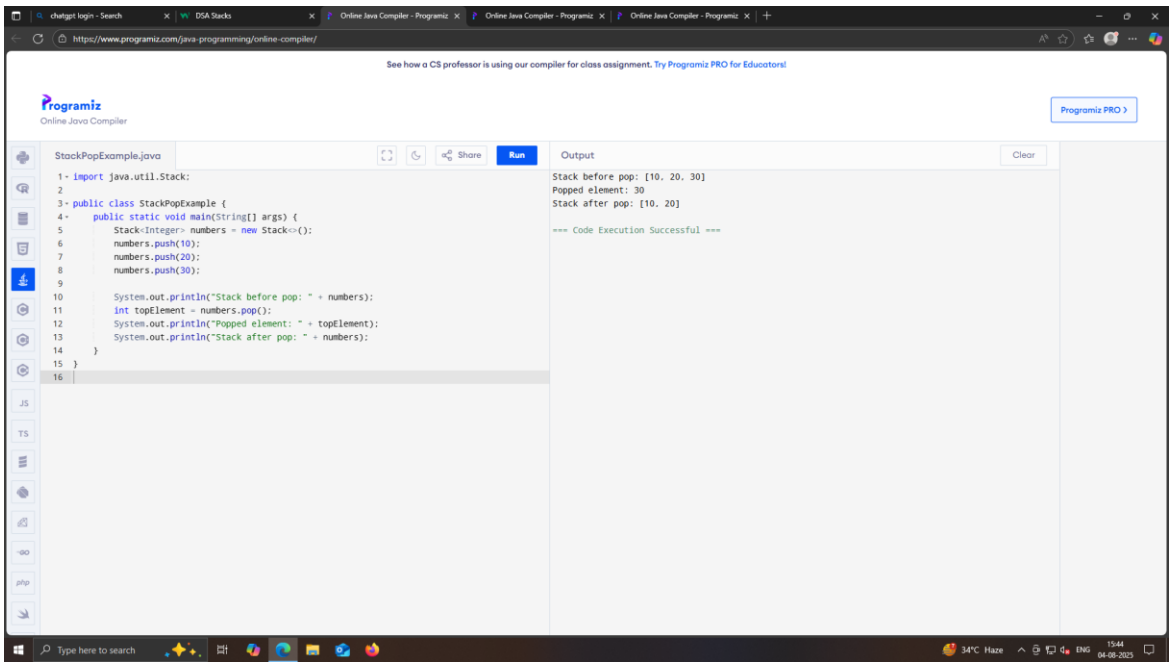
Sec. D

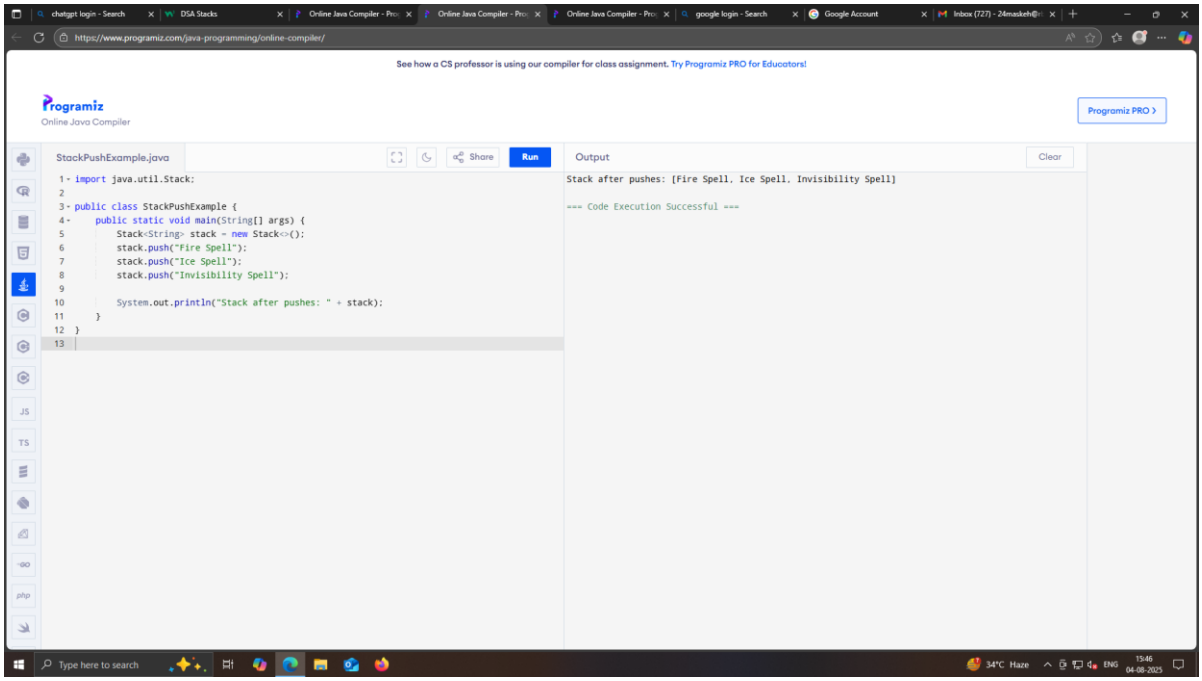
STACKS

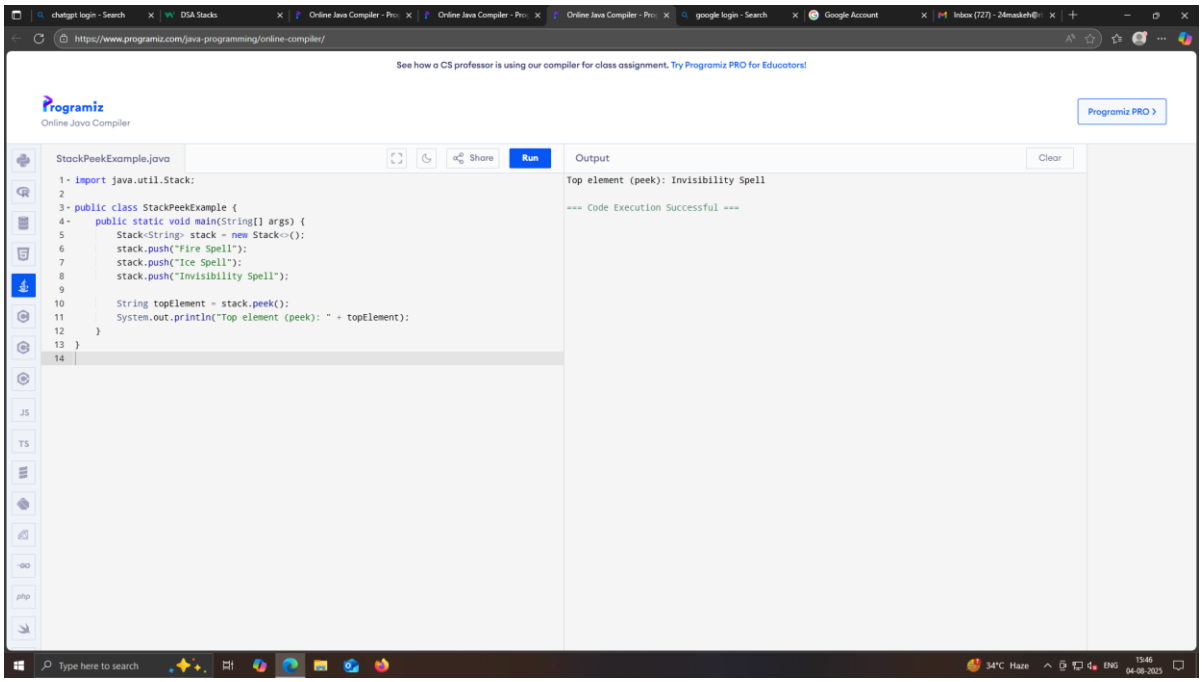
```
import java.util.Stack;

public class StackPopExample {
    public static void main(String[] args) {
        Stack<Integer> numbers = new Stack<>();
        numbers.push(10);
        numbers.push(20);
        numbers.push(30);

        System.out.println("Stack before pop: " + numbers);
        int topElement = numbers.pop();
        System.out.println("Popped element: " + topElement);
        System.out.println("Stack after pop: " + numbers);
    }
}
```

```
import java.util.Stack;

public class StackPushExample {
    public static void main(String[] args) {
        Stack<String> stack = new Stack<>();
        stack.push("Fire Spell");
        stack.push("Ice Spell");
        stack.push("Invisibility Spell");

        System.out.println("Stack after pushes: " + stack);
    }
}
```

```
import java.util.Stack;

public class StackPeekExample {
    public static void main(String[] args) {
        Stack<String> stack = new Stack<>();
        stack.push("Fire Spell");
        stack.push("Ice Spell");
        stack.push("Invisibility Spell");

        String topElement = stack.peek();
        System.out.println("Top element (peek): " + topElement);
    }
}
```




https://www.programiz.com/java-programming/online-compiler/
See how a CS professor is using our compiler for class assignment. Try Programiz PRO for Educators!
Programiz
Online Java Compiler
Programiz PRO
StackPushExample.java
Share
Run
Output
Clear
1 import java.util.Stack;
2
3 public class StackPushExample {
4 public static void main(String[] args) {
5 Stack<String> stack = new Stack<>();
6 stack.push("Fire Spell");
7 stack.push("Ice Spell");
8 stack.push("Invisibility Spell");
9
10 System.out.println("Stack after pushes: " + stack);
11 }
12 }
13
Stack after pushes: [Fire Spell, Ice Spell, Invisibility Spell]
=== Code Execution Successful ===


https://www.programiz.com/java-programming/online-compiler/
See how a CS professor is using our compiler for class assignment. Try Programiz PRO for Educators!
Programiz
Online Java Compiler
Programiz PRO
StackPeekExample.java
Share
Run
Output
Clear
1 import java.util.Stack;
2
3 public class StackPeekExample {
4 public static void main(String[] args) {
5 Stack<String> stack = new Stack<>();
6 stack.push("Fire Spell");
7 stack.push("Ice Spell");
8 stack.push("Invisibility Spell");
9
10 String topElement = stack.peek();
11 System.out.println("Top element (peek): " + topElement);
12 }
13 }
14
Top element (peek): Invisibility Spell
=== Code Execution Successful ===

```
import java.util.Scanner;

public class UserDefinedStackExample {

    static int maxSize;
    static int[] stackArray;
    static int top = -1;

    // Initialize stack with user-defined size
    public static void initializeStack(int size) {
        maxSize = size;
        stackArray = new int[maxSize];
        top = -1;
    }

    public static boolean isFull() {
        return top == maxSize - 1;
    }

    public static boolean isEmpty() {
        return top == -1;
    }

    public static void push(int value) {
        if (!isFull()) {
            stackArray[++top] = value;
            System.out.println(value + " pushed into stack");
        } else {
            System.out.println("Stack is full. Cannot push " + value);
        }
    }

    public static int pop() {
        if (!isEmpty()) {
            int popped = stackArray[top--];
            System.out.println(popped + " popped from stack");
            return popped;
        } else {
            System.out.println("Stack is empty. Cannot pop");
            return -1;
        }
    }

    public static int peek() {
        if (!isEmpty()) {
            int topValue = stackArray[top];
            System.out.println("Top element is " + topValue);
            return topValue;
        } else {
            System.out.println("Stack is empty. Nothing to peek");
```

```
            return -1;
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter stack size: ");
        int size = sc.nextInt();
        initializeStack(size);

        while (true) {
            System.out.println("\nChoose operation: 1-Push 2-Pop 3-Peek 4-Exit");
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter value to push: ");
                    int val = sc.nextInt();
                    push(val);
                    break;
                case 2:
                    pop();
                    break;
                case 3:
                    peek();
                    break;
                case 4:
                    System.out.println("Exiting...");
                    sc.close();
                    System.exit(0);
                default:
                    System.out.println("Invalid choice, try again.");
            }
        }
    }
}
```