Activity 1 – Stack Programs

Stack Using Array

```
import java.util.Scanner;

class StackArray {
    private int max = 5;
    private int top = -1;
    private int[] stack = new int[max];

    void push(int value) {
        if (top == max - 1) {
            System.out.println("Stack Overflow!");
            return;
        }
        stack[++top] = value;
        System.out.println(value + " pushed into stack");
    }

    void pop() {
        if (top == -1) {
            System.out.println("Stack Underflow!");
            return;
        }
        System.out.println(stack[top--] + " popped");
    }

    void peek() {
        if (top == -1) {
            System.out.println("Stack Empty");
            return;
        }
        System.out.println("Top element: " + stack[top]);
    }

    void display() {
        if (top == -1) {
            System.out.println("Stack Empty");
            return;
        }
        System.out.print("Stack: ");
        for (int i = top; i >= 0; i--)
            System.out.print(stack[i] + " ");
        System.out.println();
    }

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

        while (true) {
            System.out.println("\n1.Push 2.Pop 3.Peek 4.Display 5.Exit");
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter value: ");
                    s.push(sc.nextInt());
                    break;
                case 2:
                    s.pop();
                    break;
                case 3:
                    s.peek();
                    break;
                case 4:
                    s.display();
                    break;
                case 5:
```

```
                    System.exit(0);
                }
            }
        }
    }
```

Balanced Parentheses Checker

```
import java.util.Stack;
import java.util.Scanner;

public class BalancedChecker {

    public static boolean isBalanced(String expr) {
        Stack<Character> st = new Stack<>();

        for (char ch : expr.toCharArray()) {

            if (ch == '(' || ch == '{' || ch == '[')
                st.push(ch);

            else if (ch == ')' || ch == '}' || ch == ']') {
                if (st.isEmpty()) return false;

                char top = st.pop();

                if ((ch == ')' && top != '(') ||
                    (ch == '}' && top != '{') ||
                    (ch == ']' && top != '['))
                    return false;
                }
            }
            return st.isEmpty();
        }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter expression: ");
        String exp = sc.nextLine();

        if (isBalanced(exp))
            System.out.println("Balanced");
        else
            System.out.println("Not Balanced");
    }
}
```