Task 4
Stack Sorting In-Place
You must write a function to sort a stack such that the smallest items are on the top. You can use an additional temporary stack, but you may not copy the elements into any other data structure such as an array. The stack supports the following operations: push, pop, peek, and isEmpty.
ANS:

```
package Assigmentday12.com;
import java.util.Stack;
public class Task4 {
  public static void sortStack(Stack<Integer> stack) {
     Stack<Integer> tempStack = new Stack<>();
     while (!stack.isEmpty()) {
        int temp = stack.pop();
        while (!tempStack.isEmpty() && tempStack.peek() < temp) {
           stack.push(tempStack.pop());
        }
        tempStack.push(temp);
     }
     // Move elements from tempStack back to stack
     while (!tempStack.isEmpty()) {
        stack.push(tempStack.pop());
     }
  }
  public static void main(String[] args) {
     Stack<Integer> stack = new Stack<>();
     stack.push(4);
     stack.push(1);
     stack.push(3);
     stack.push(2);
     System.out.println("Original Stack: " + stack);
     sortStack(stack);
     System.out.println("Sorted Stack: " + stack);
  }
}
```

OUTPUT:

```
Original Stack: [4, 1, 3, 2]
Sorted Stack: [1, 2, 3, 4]
```