

Credit Name: Chapter13  
Assignment Name: ReverseList

How has your program changed from planning to coding to now? Please explain?

At first, the program was planned to show how a stack can be used to reverse a list of integers entered by the user. The initial plan included creating a ReverseList class with a stack to manage the elements and a test class to prompt the user for input. Methods like push(), pop(), and isEmpty() from the stack implementation were used to reverse the order of elements. The program was supposed to handle a maximum of 10 numbers and terminate when the user entered 999.

### 1. Infinite Input Loop

Problem: The program continued prompting the user for input even after entering 999.

Fix: I added a break statement when 999 was entered to exit the loop and terminate the input process.

```
// Before
while (stack.size() < 10) {
    num = input.nextInt();
    if (num == 999) {
    }
    stack.push(num);
}

// After
while (stack.size() < 10) {
    num = input.nextInt();
    if (num == 999) {
        break;
    }
    stack.push(num);
}
```

## 2. Calling pop() on an Empty Stack

Problem: The program threw an EmptyStackException when attempting to reverse an empty list.

Fix: I added a condition to check if the stack was not empty before calling pop().

```
// Before
while (true) {
    System.out.print(stack.pop() + " ");
}
// After
while (!stack.isEmpty()) {
    System.out.print(stack.pop() + " ");
}
```

## 3. Missing Input Validation

Problem: The program threw an InputMismatchException when the user entered invalid (non-integer) input.

Fix: I used a try-catch block to handle invalid input and prompt the user to enter a valid number.

```
// Before fix
int num = input.nextInt();

// After fix
int num;
while (true) {
    try {
        num = input.nextInt();
        break;
    } catch (Exception e) {
        System.out.println("Invalid input. Please enter an integer.");
        input.next();
    }
}
```

#### 4. Stack Overflow Due to Too Many Items

Problem: The stack overflowed if more than 10 items were entered.

Fix: I added a condition to ensure the stack size does not exceed 10 elements.

```
// Before
while (true) {
    stack.push(num);
}
// After
while (stack.size() < 10) {
    num = input.nextInt();
    if (num == 999) {
        break;
    }
    stack.push(num);
}
```

Now, the program works properly. It takes user input, validates it, and correctly reverses the entered numbers. These adjustments improved its reliability and ensured a smoother user experience.