→ STACK

→ It follows LIFO (Last in First out) principle

→ Implementing a ~~queue~~ Stack

STACK <INT> S;

→ Adding in a ~~queue~~ Stack

S. PUSH (2);
S. PUSH (3);
S. PUSH (4);

→ Emptying a of stack

```
WHILE (!S.EMPTY ())
{
    → Accessing top element
    COUT << S.TOP() << ENDL;
    → Removing top element
    S.POP();
}
```