

Data Structures in C++

CMPE226- Data Structures

Stacks

STACKS

- A data structure in which elements are added & removed only from one end.
- Last In First Out (LIFO) structure

The last element inserted will be the first to be retrieved.
- Implemented as array or linked list
 - Arrays: limited number of elements
 - Linked lists: allow dynamic element addition

Stacks

- Data structure
 - Elements added, removed from one end only
 - Last In First Out (LIFO)

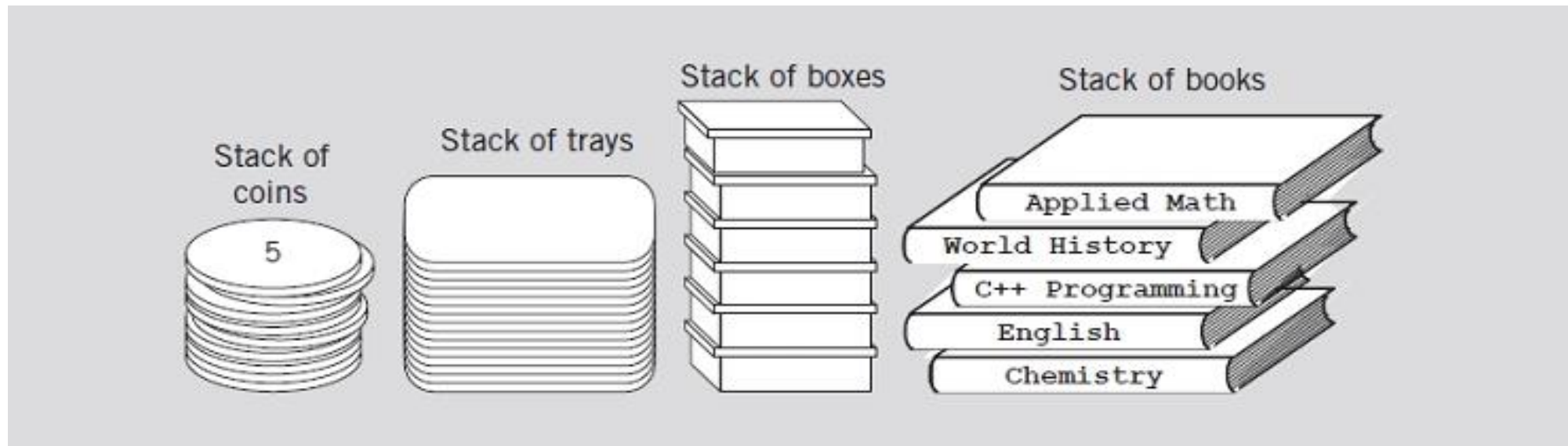


FIGURE 7-1 Various examples of stacks

STACKS

Stack ADT Definition

- Initialize: Initialize stack to empty stack
- Destroy: remove all elements
- isEmpty: return true if stack is empty, false otherwise
- isFull: return false if stack is full, true otherwise
- Push: add a new element to the top
- Pop: remove and return top element
- Top: return top element (not remove only return)

Stacks (cont'd.)

- **push operation**
 - Add element onto the stack
- **top operation**
 - Retrieve top element of the stack
- **pop operation**
 - Remove top element from the stack

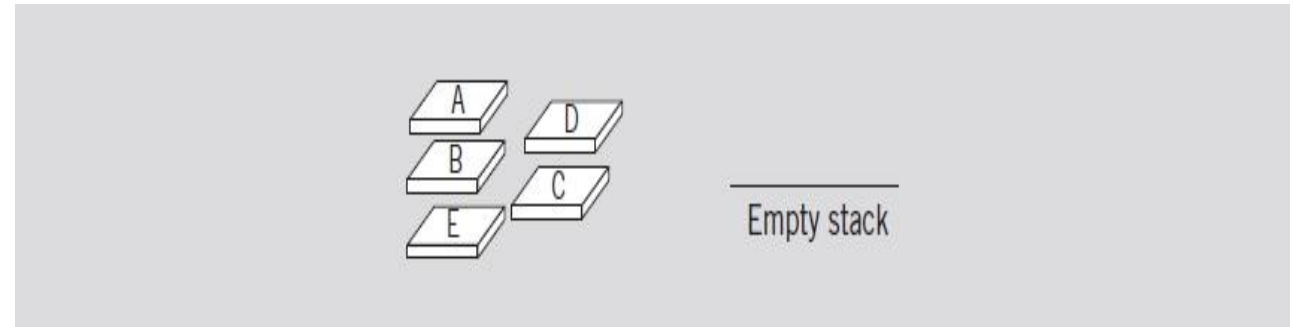


FIGURE 7-2 Empty stack

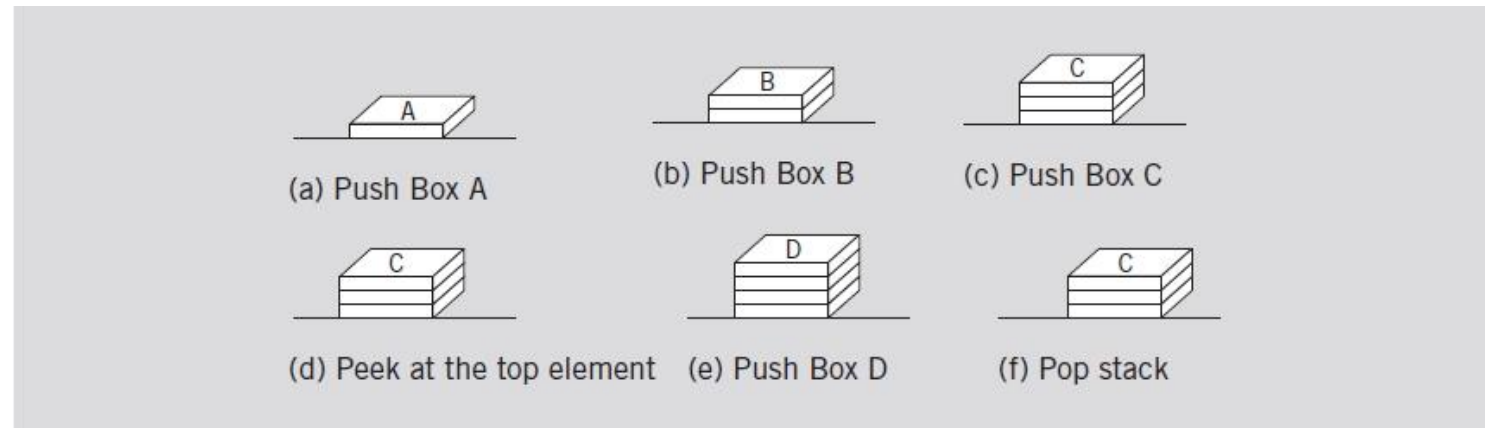
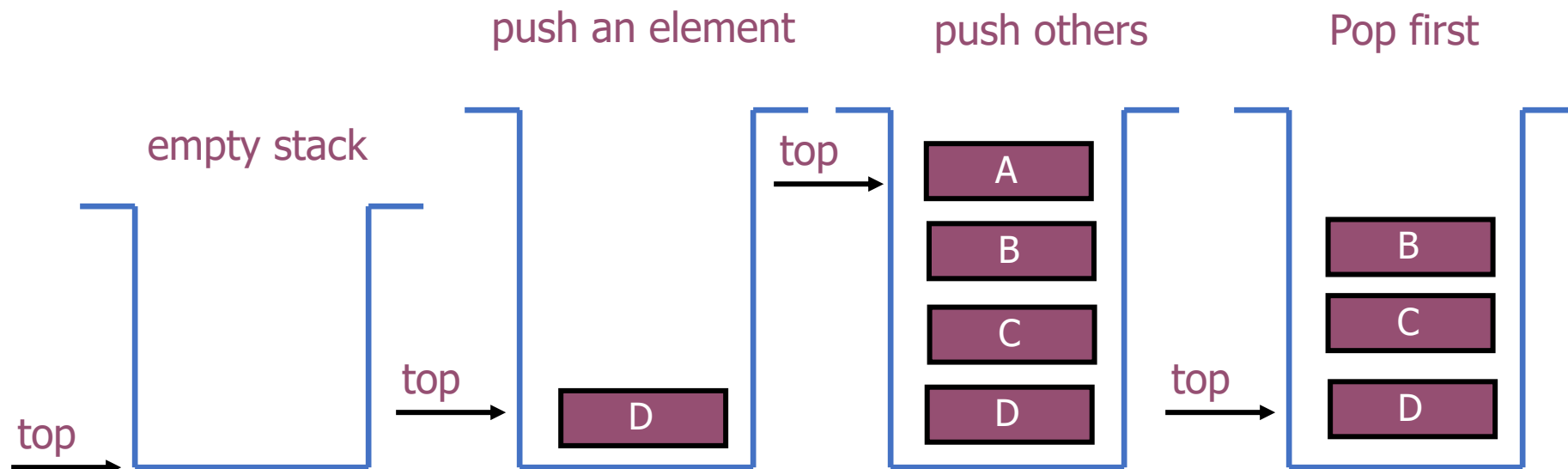


FIGURE 7-3 Stack operations

Example: Push and Pop

- Push: Add an element to the top of the stack
- Pop: Remove the element at the top of the stack
- **Exp.** If the characters 'D', 'C', 'B', 'A' is placed in a stack (in that order), and then removed one at a time, in what order will they be removed?



Example: Draw the final version of stack after following commands executed.

Initialize s.

Push(2);

Push(3);

Push(5);

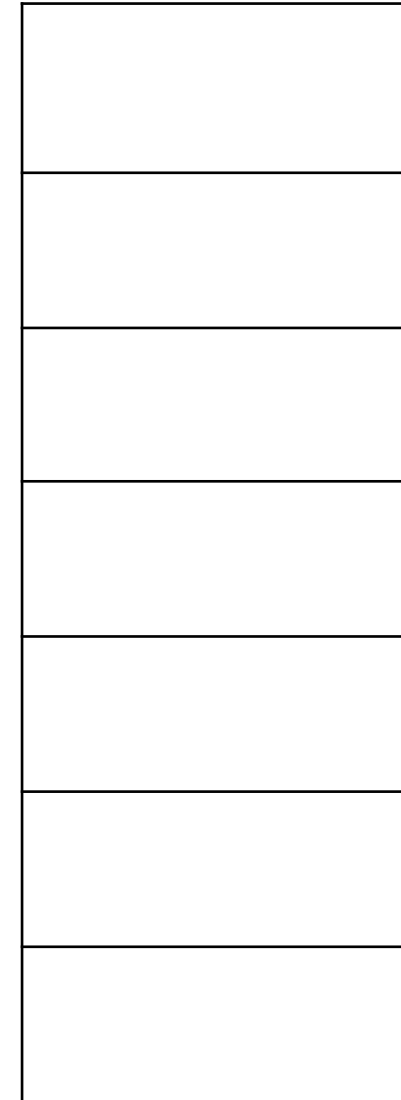
X=pop();

Y= pop();

Push(6);

X=pop();

Push(y);



Implementation of Stacks- with Arrays

```
template <class T>
class Stack {
protected:
    T *arr;//array hold elements
    int top;//index of empty space after topmost element
    int size;//size of array
public:
    Stack(int size = 100);
    bool isEmpty();
    bool isFull();
    void destroy();
    void push(T &);
    void copy(Stack<T> &);
    T pop();
    T top();//retrieve top element
    ~Stack();
};
```


Implementation of Stacks- with Arrays

```
template <class T>
Stack <T>::Stack(int stackSize) {
    size = stackSize;
    arr = new T[size];
    top = 0;
}
template <class T>
Stack<T>::~~Stack() {
    delete[] arr;
    size = 0;
}
template <class T>
bool Stack<T>::isEmpty() {
    return (top == 0);
}
template <class T>
bool Stack<T>::isFull() {
    return (top == size);
}
```

Implementation of Stacks- with Arrays

```
template <class T>
void Stack<T>::destroy() {
    top = 0;
}
template <class T>
void Stack<T>::push(T &data) {
    if (!isFull()) {
        arr[top++] = data;
    }else {
        std::cout << "Stack Full" << std::endl;
    }
}
template <class T>
T Stack<T>::pop() {
    assert(!isEmpty());
    // If function evaluates to false, abort (FOR DEBUGGING)

    return arr[--top];
}
```

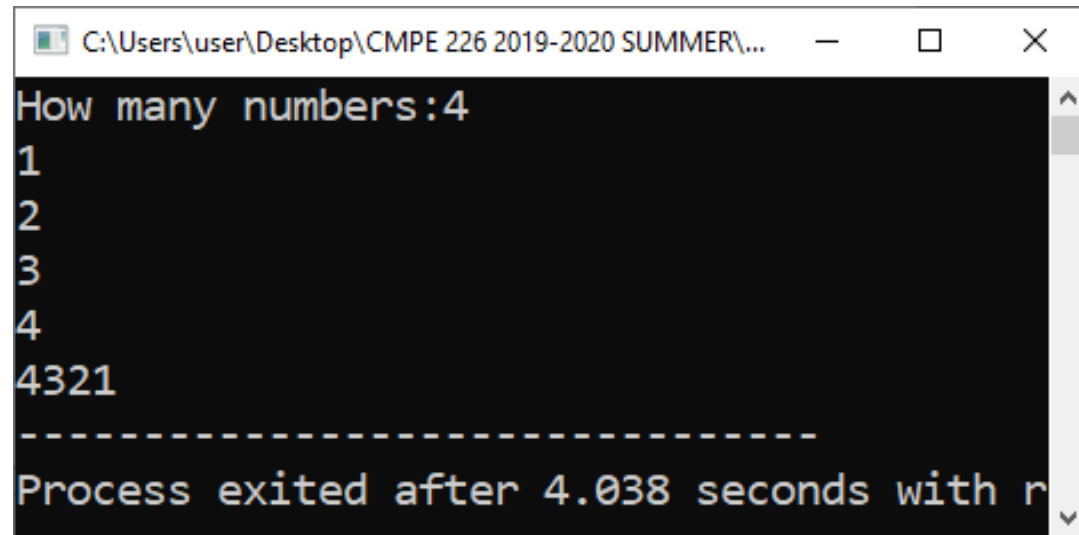
Implementation of Stacks- with Arrays

```
template <class T>
T Stack<T>::topData() {
    assert(!isEmpty());
    // If function evaluates to false, abort (FOR DEBUGGING)
    return arr[top-1];
}
```

```
template <class T>
void Stack<T>::copy(Stack<T> &st) {
    delete[] arr;
    size = st.size;
    top = st.top;
    arr = new T[size];
    for (int n = 0; n < top; n++) {
        arr[n] = st.arr[n];
    }
}
```

Example

- Input an integer n from user and input n integers. Output them in reverse order. Use stack.h file.

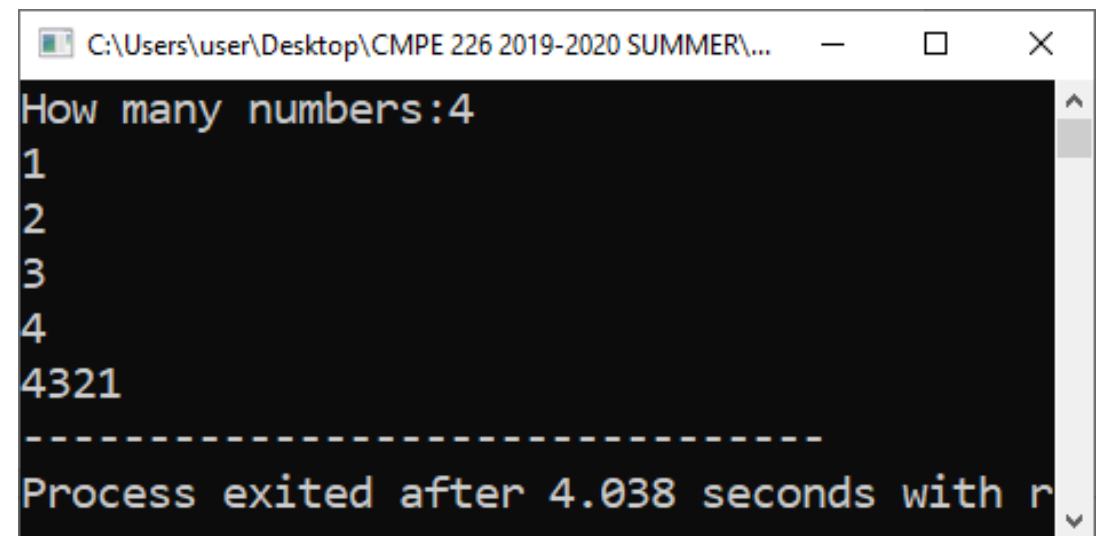


```
C:\Users\user\Desktop\CMPE 226 2019-2020 SUMMER\...
How many numbers:4
1
2
3
4
4321
-----
Process exited after 4.038 seconds with r
```

Example

- Input an integer n from user and input n integers. Output them in reverse order. Use stack.h header file.

```
#include<iostream>
#include "stack.h"
using namespace std;
main(){
    Stack<int> numbers(10);
    int n,x;
    cout<<"How many numbers:";
    cin>>n;
    for(int i=0;i<n;i++){
        cin>>x;
        numbers.push(x);
    }
    while(!numbers.isEmpty()){
        cout<<numbers.pop();
    }
}
```



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\user\Desktop\CMPE 226 2019-2020 SUMMER\... The window contains the following text: "How many numbers:4", followed by the numbers "1", "2", "3", and "4" on separate lines. Below these is the output "4321". A dashed line separates the input from the output. At the bottom, it says "Process exited after 4.038 seconds with r".

References

- CMPE226- Data Structures Lecture Notes by Cigdem Turhan
- Data Structures Using C++, D.S. Malik, Thomson Course Technology, 2nd Edition.
- Lecture Slides by Huamin Qu, The Hong Kong University of Science and Technology (2005)