SNS COLLEGE OF TECHNOLOGY

Coimbatore-35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A+’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE NAME : COURSE NAME : 23ITT101&PROGRAMMING IN C AND DATA STRUCTURES

I YEAR/ II SEMESTER

UNIT – IV STACK AND QUEUE

Topic: Stack ADT

Abstarct Data Type-What is Data Type

Two important things about data types

1.Defines a certain domain of values

2.Defines operations allowed on those values

Example

int type

Takes only integer values

Operations : addition,subtraction,multiplication,bitwise operation etc

Introduction- User defined Data type

- The operations and values of user defined data types are not specified in the language itself but is specified by the user
- Example :Structure,Union and enumeration

```
 Struct point{
             int x;
             int y;
                  }
```

Defining our own type by combining other data types

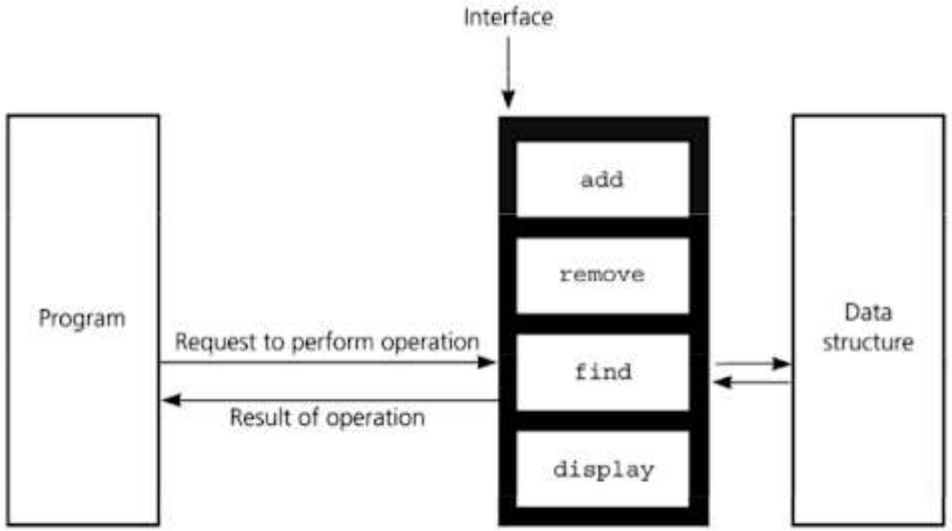
Abstract Data type

ADT are like user defined data types which defines operations on values using functions without specifying what is there inside the functions and how the operations are performed

Example Stack ADT

Operations

- ➢Initialize() initializing it to be Empty
- ➢Push()-Insert an element into stack
- ➢Pop()-Delete an element from the stack
- ➢isEmpty()-Checks if stack is empty
- ➢isFull()-ckecks if stack is full



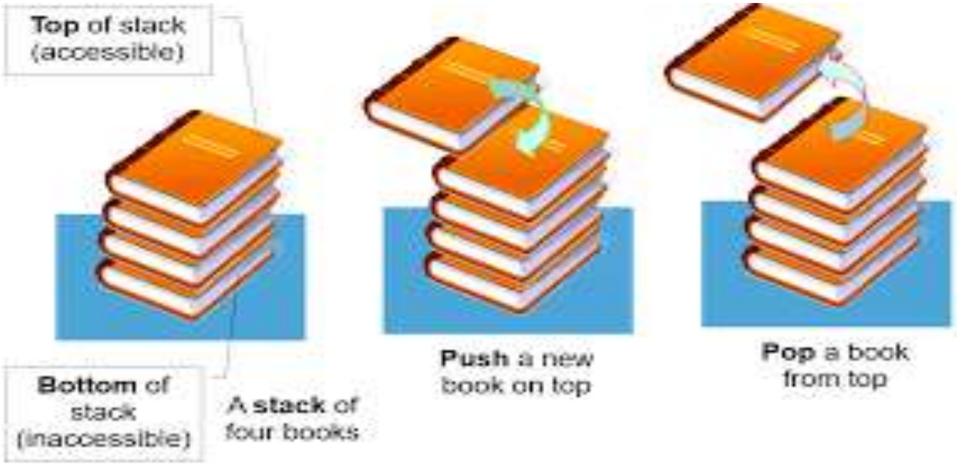
Abstract Data type

- ADT as a black box which hides the inner structure and design of the data type from the user
- Multiple ways to implement data type
- Example :
- Stack ADT can be implemented using arrays or linked lists
- The program which uses data structure is called a client program.It has access to the ADT ie Interface
- The program which implements the data structure is known as the implementation

Stack ADT

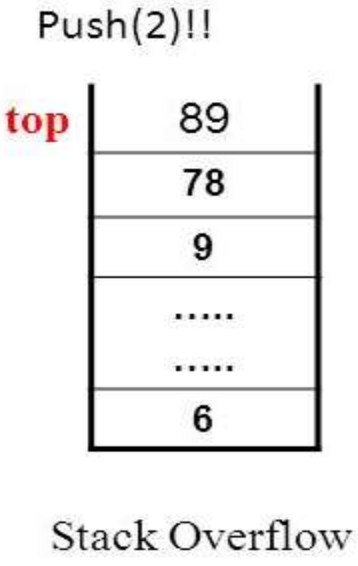
- ➢Linear Data Structure
- ➢Elements are arranged in sequential
- ➢Follows LIFO-Last In First Out mechanism
- ➢Example-Plates in tray
- ➢Arranging the coins ,Books
- ➢Push-Inserting the element into the stack
- ➢Pop-Deleting the element from the stack
- ➢Top-top of stack

top=-1



Implemented in two ways
Using Arrays
Using Linked List

Stack



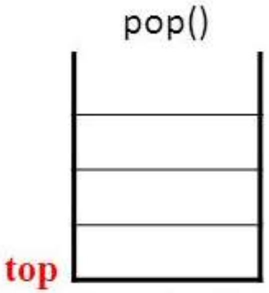
Stack Overflow

Overflow condition(push)

- Stack-5 elements
- Insert 6 th element
- Inserting an element to the stack which is already having maximum elements

Underflow Condition

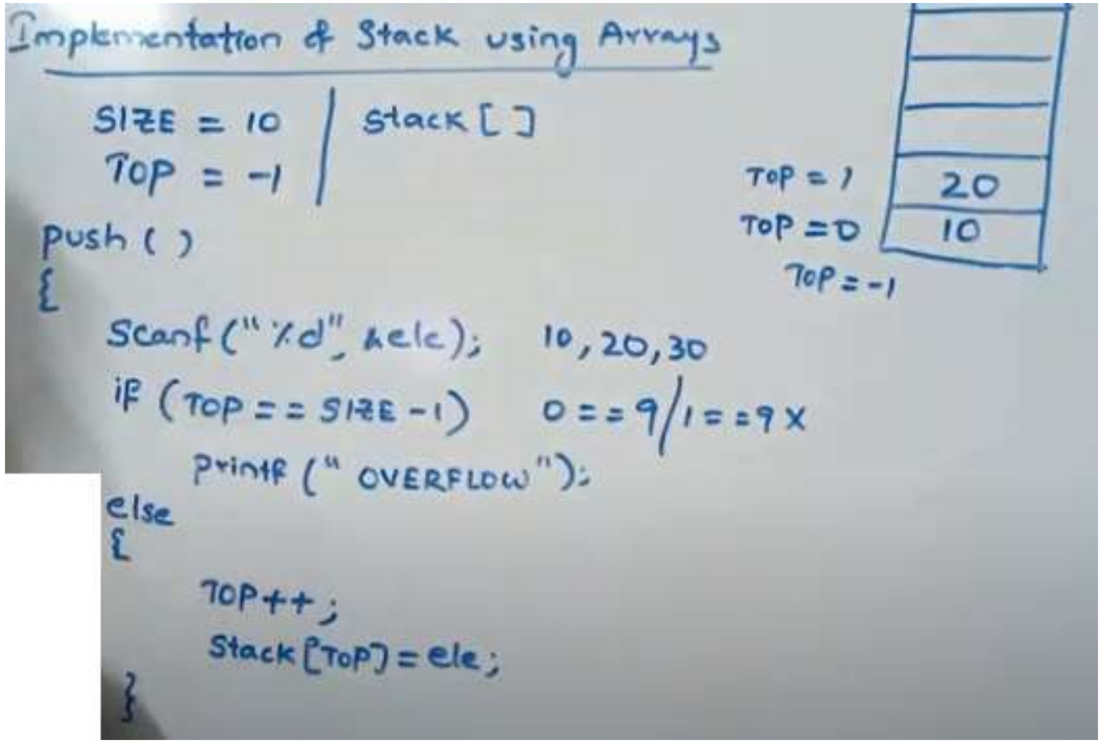
Deleting an element from the empty stack



Stack Underflow

Implementation of Stack Using Arrays

Push operation

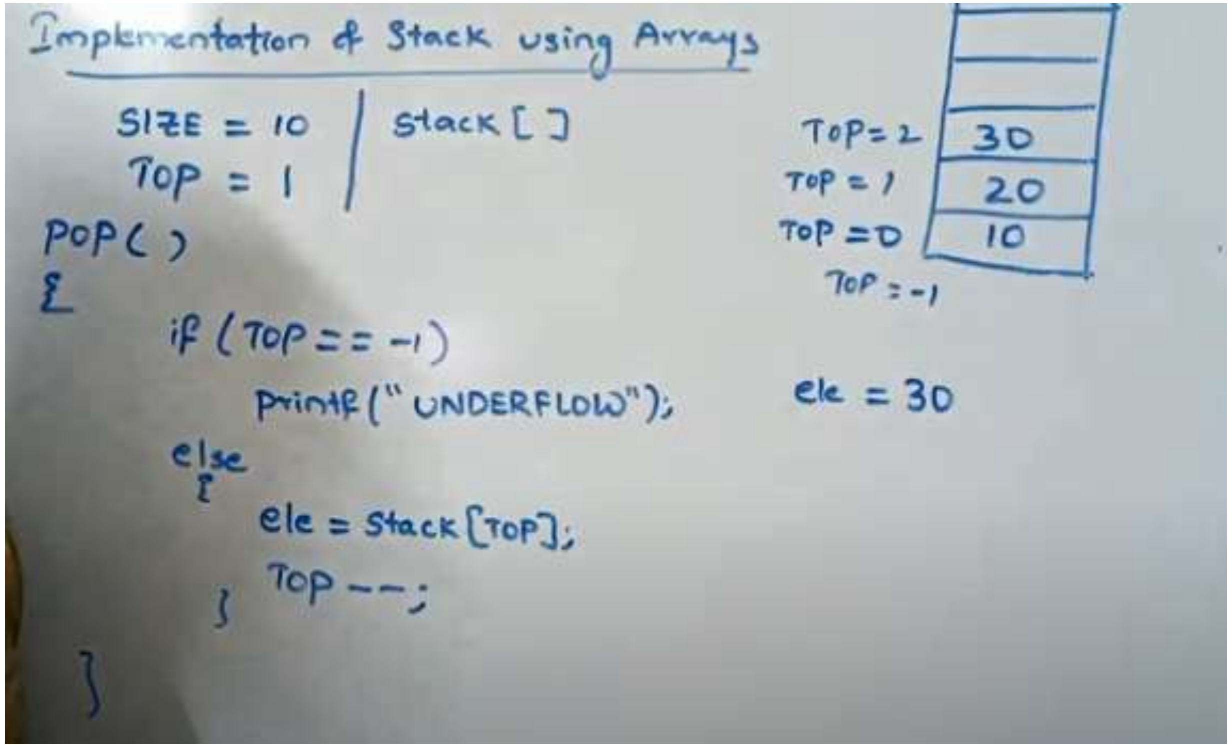


Implementation of Stack using Arrays

```
SIZE = 10  |  stack[ ]
Top = -1   |

push ( )
{
    Scanf("%d", &ele);         10,20,30
    if (Top == SIZE-1)         0==9 / 1==9 X
        Printf(" OVERFLOW");
    else
    {
        TOP++;
        Stack[TOP] = ele;
    }
```

```
TOP = 1   | 20 |
TOP = 0   | 10 |
TOP = -1
```

Pop operation



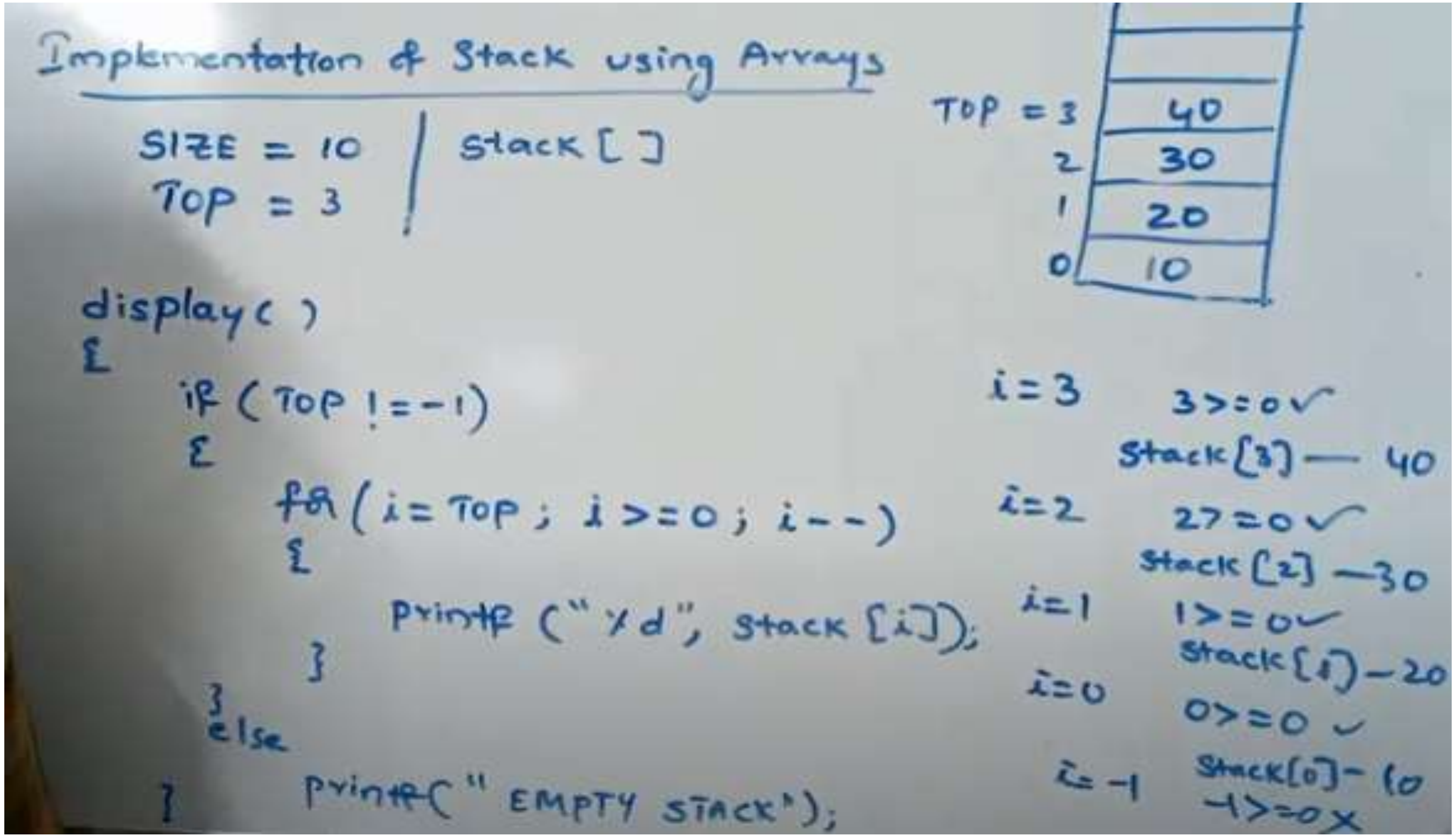
Implementation of Stack using Arrays

```
SIZE = 10  |  stack[ ]
Top = 1    |

POP( )
{
    if (TOP == -1)
        Printf("UNDERFLOW");
    else
    {
        ele = Stack[TOP];
        Top --;
    }
}
```

```
TOP = 2   | 30 |
TOP = 1   | 20 |
TOP = 0   | 10 |
TOP = -1

ele = 30
```

Implementation of Stack Using Arrays

Display Function



Implementation of Stack using Arrays

SIZE = 10 | stack[ ]
TOP = 3

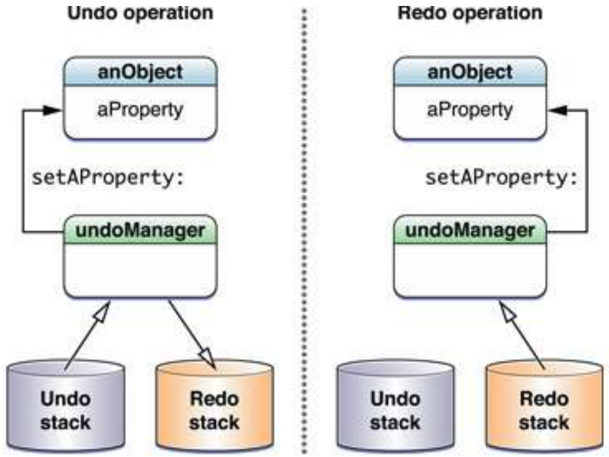
```
display( )
{
    if (TOP != -1)
    {
        for (i= TOP ; i >=0 ; i--)
        {
            printf ("%d", stack [i]);
        }
    }
    else
        printf(" EMPTY STACK");
}
```

TOP = 3	40
2	30
1	20
0	10

i=3   3>=0 ✓
stack[3] — 40
i=2   2>=0 ✓
stack[2] — 30
i=1   1>=0 ✓
stack[1] — 20
i=0   0>=0 ✓
stack[0] — 10
i= -1   -1>=0 ✗

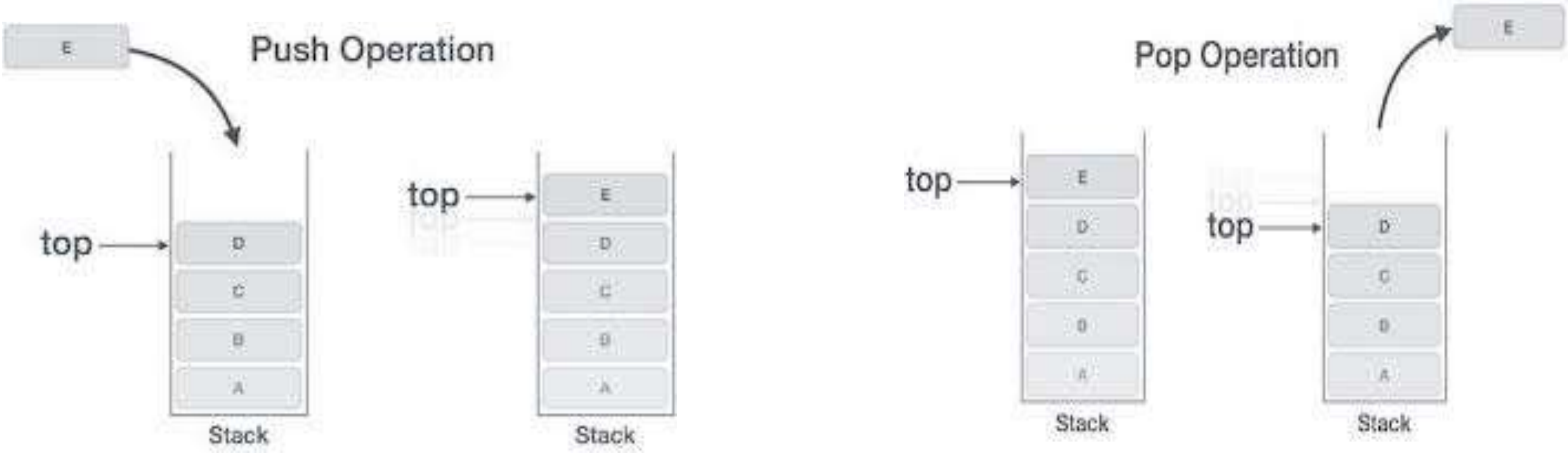
Applications of Stack

- **Evaluating arithmetic Expression**
  - **Infix to Postfix Conversion**
  - **Evaluating the Postfix Expression**
- Balancing the Symbols
- Undo/Redo
- Reverse a string
- Towers of Hanoi
- Function calls
- 8 Queen Problem



Assessment

1.How to do Push and POP Operations



2.Tell any two applications of Stack

References

1. M. A. Weiss, “Data Structures and Algorithm Analysis in C”, Pearson Education, 2nd Edition, 2002.
2. A. V. Aho, J. E. Hopcroft and J. D. Ullman, “Data Structures and Algorithms”, Pearson Education, 2nd Edition, 2007
3. Ashok Kamthane, " Data Structures Using C ", Pearson Education, 2nd Edition, 2012.
4. Sahni Horowitz, “Fundamentals of Data Structures in C”Universities Press; Second edition 2008

Thank You