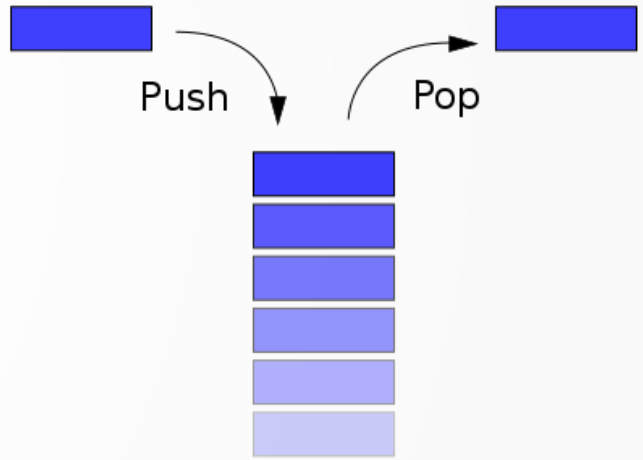
Data Structures – Lesson 3: Stack

B.Sc. 3rd Semester, Paper C5

Paulami Basu Ray
Assistant Professor
Department of Computer Science & Applications
Prabhat Kumar College, Contai

Stack: definition

- A stack is an ordered list in which ***insertion*** & **deletion** are done at one end, called ***top***. The last element inserted is the first one to be deleted. Hence, it is called the ***Last in First out (LIFO)*** or ***First in Last out (FILO)*** list.
- When an element is inserted in a stack, the concept is called ***push***.
- When an element is deleted in a stack, the concept is called ***pop***.



Some real life examples of Stack

A stack of cafeteria trays

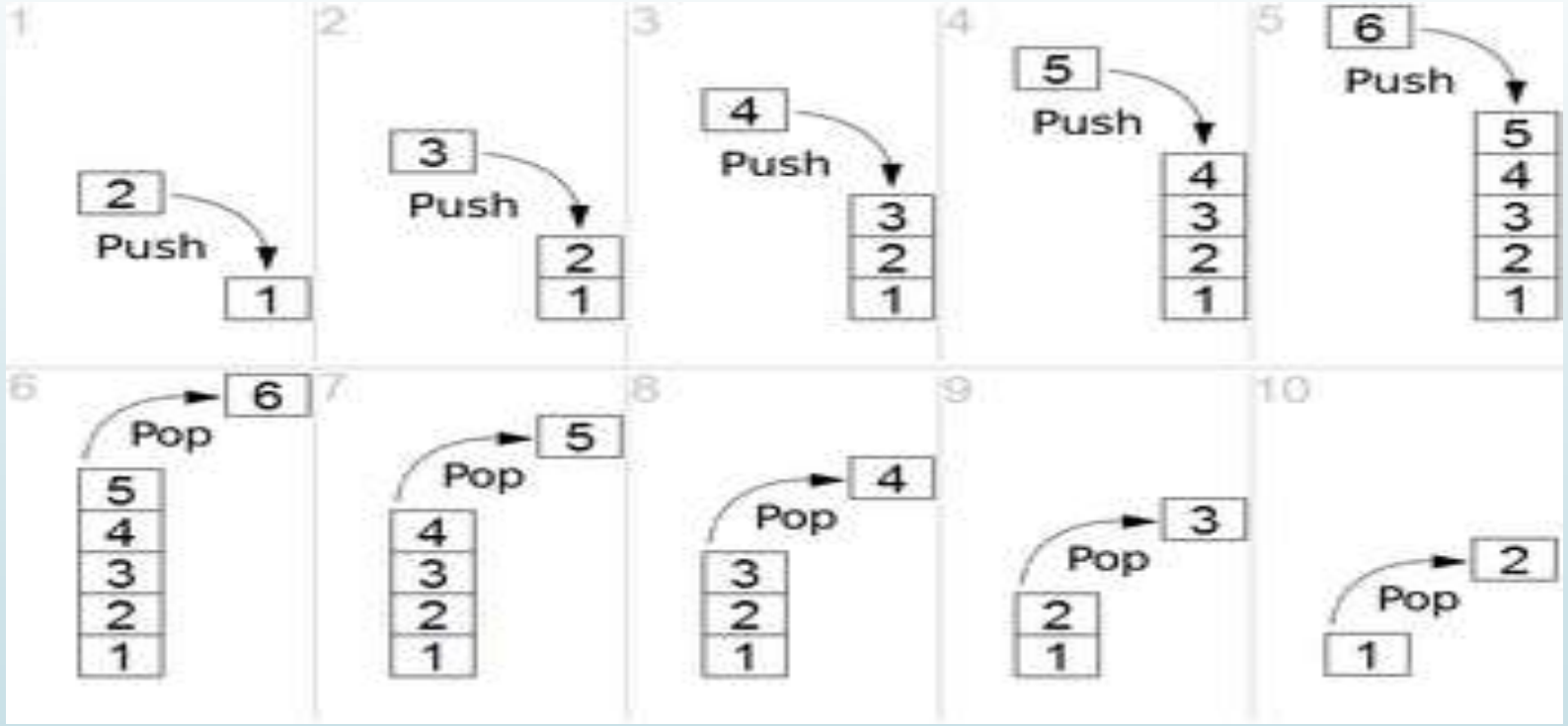
A stack of pennies

A stack of shoe boxes

A stack of neatly folded shirts

Some more examples

Stack Operations



C Program for Stack implementation

```
#include<stdio.h>
#include<process.h>
#include<stdlib.h>
#define MAX 5 //Maximum number of elements that can be stored

int top=-1,stack[MAX];
void push();
void pop();
void display();

void main()
{
        int ch;
        while(1) //infinite loop, will end when choice will be 4
        {
        printf("\n*** Stack Menu ***");
        printf("\n\n1.Push\n2.Pop\n3.Display\n4.Exit");
        printf("\n\nEnter your choice(1-4):");
        scanf("%d",&ch);
```

```
switch(ch)
{
        case 1: push();
        break;
        case 2: pop();
        break;
        case 3: display();
        break;
        case 4: exit(0);
        default: printf("\nWrong Choice!!");
}
}
}
```

```
void push()
{
        int val;
        if(top==MAX-1)
        {
                printf("\nStack is full!!");
        }
        else
        {
                printf("\nEnter element to push:");
                scanf("%d",&val);
                top=top+1;
                stack[top]=val;
        }
}
```

```
void pop()
{
    if(top==-1)
    {
    printf("\nStack is empty!!");
    }
    else
    {
    printf("\nDeleted element is %d",stack[top]);
    top=top-1;
    }
}
```

```
void display()
{
        int i;
        if(top==-1)
        {
                printf("\nStack is empty!!");
        }
        else
        {
                printf("\nStack is...\n");
                for(i=top;i>=0;--i)
                printf("%d\n",stack[i]);
        }
}
```