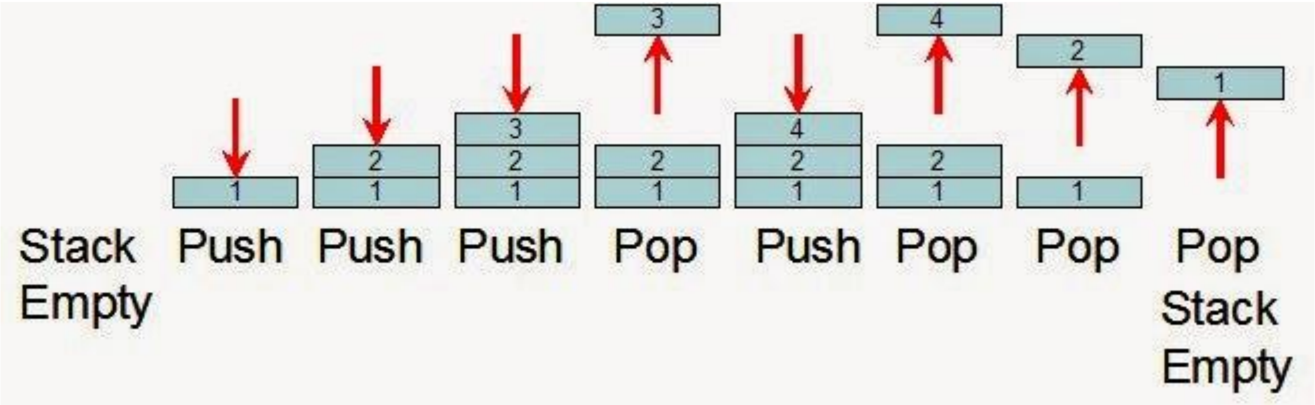
ARRAY-BASED STACK

Lab 2

AGENDA

- ❑ **General Stack Concept.**
- ❑ **Array stack class and constructors.**
- ❑ **TASK #1: Push , pop & expand functions.**
- ❑ **TASK #2: Stack Application.**

GENERAL STACK CONCEPT



ARRAY-BASED STACK CLASS
Header File & Constructors

Data Members:

1. Pointer to stack array.
2. Number of elements in stack.
3. Total size of stack array.

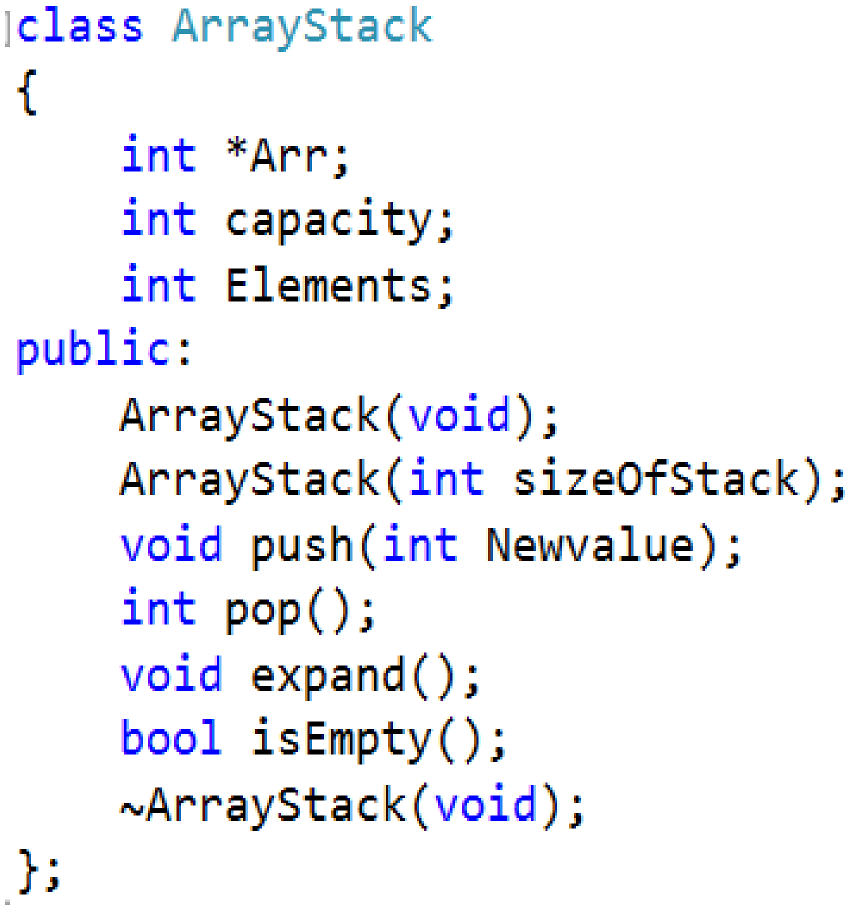
Write implementation for class Array Stack:

Methods:

1. Constructors.
2. Push.
3. Pop.
4. Expand.

ARRAY-BASED STACK CLASS
Header File

ArrayStack class (Header File)

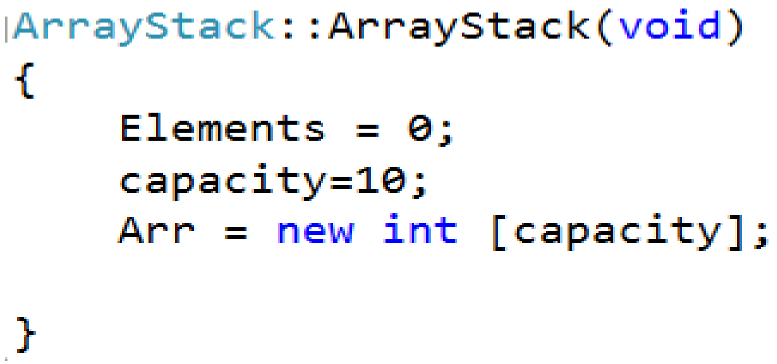


```
class ArrayStack
{
	int *Arr;
	int capacity;
	int Elements;
public:
	ArrayStack(void);
	ArrayStack(int sizeOfStack);
	void push(int Newvalue);
	int pop();
	void expand();
	bool isEmpty();
	~ArrayStack(void);
};
```

ARRAY-BASED STACK CLASS CONSTRUCTORS

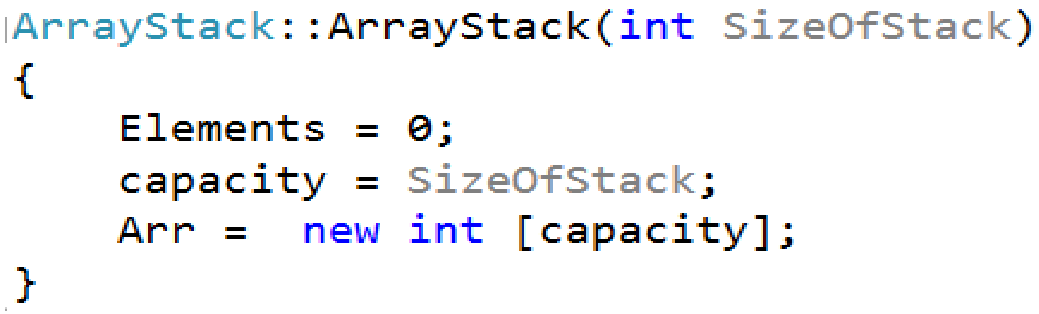
1. ArrayStack class (.CppFile)

- **Constructors**



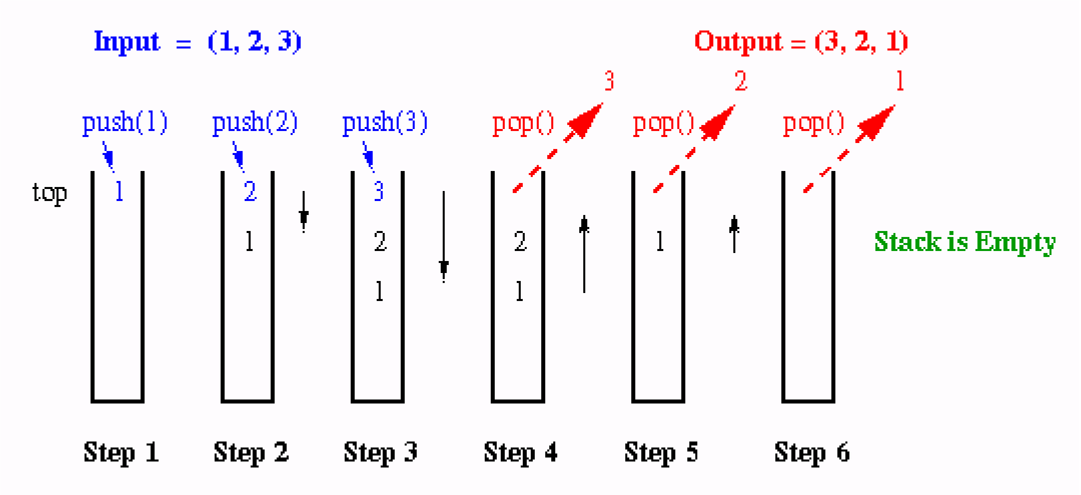
```
ArrayStack::ArrayStack(void)
{
    Elements = 0;
    capacity=10;
    Arr = new int [capacity];

}
```



```
ArrayStack::ArrayStack(int SizeOfStack)
{
    Elements = 0;
    capacity = SizeOfStack;
    Arr =  new int [capacity];
}
```

TASK 1: IMPLEMENT PUSH , POP & EXPAND

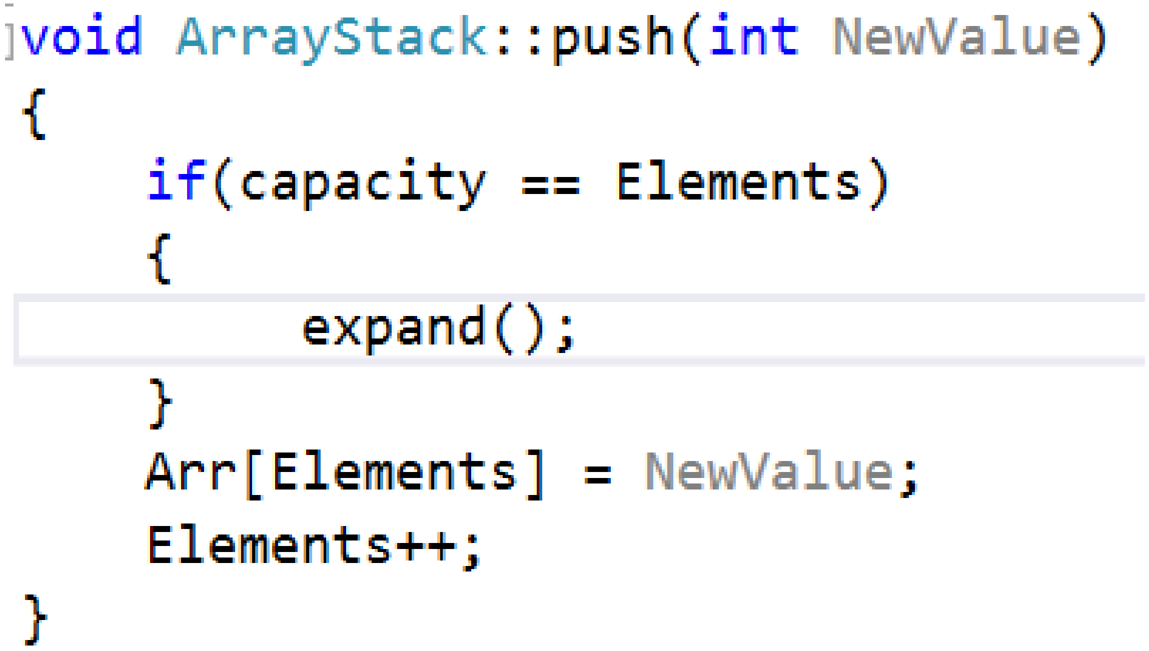


TASK 1: IMPLEMENT PUSH , POP & EXPAND *“SOLUTION”*

10 minutes

ArrayStack class (.CppFile)

- **Push**



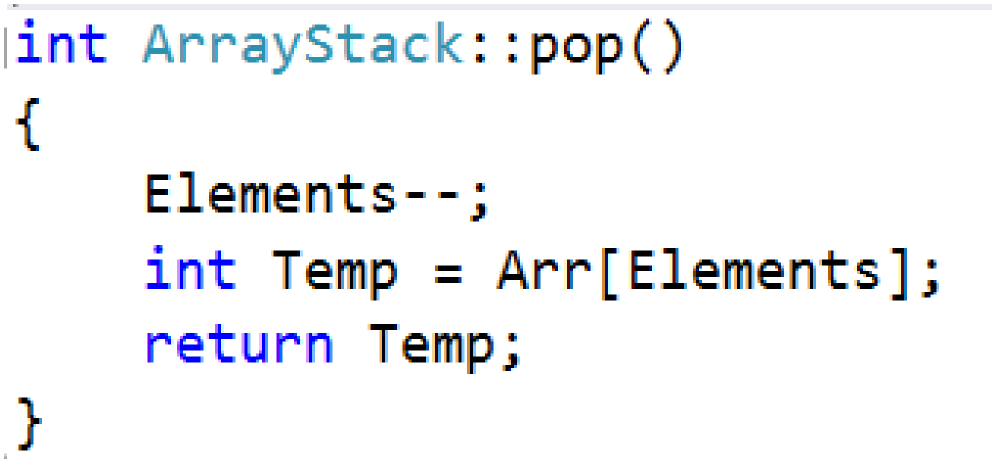
```
void ArrayStack::push(int NewValue)
{
    if(capacity == Elements)
    {
        expand();
    }
    Arr[Elements] = NewValue;
    Elements++;
}
```

TASK 1: IMPLEMENT PUSH , POP & EXPAND *“SOLUTION”*

10 minutes

ArrayStack class (.CppFile)

- **Pop**



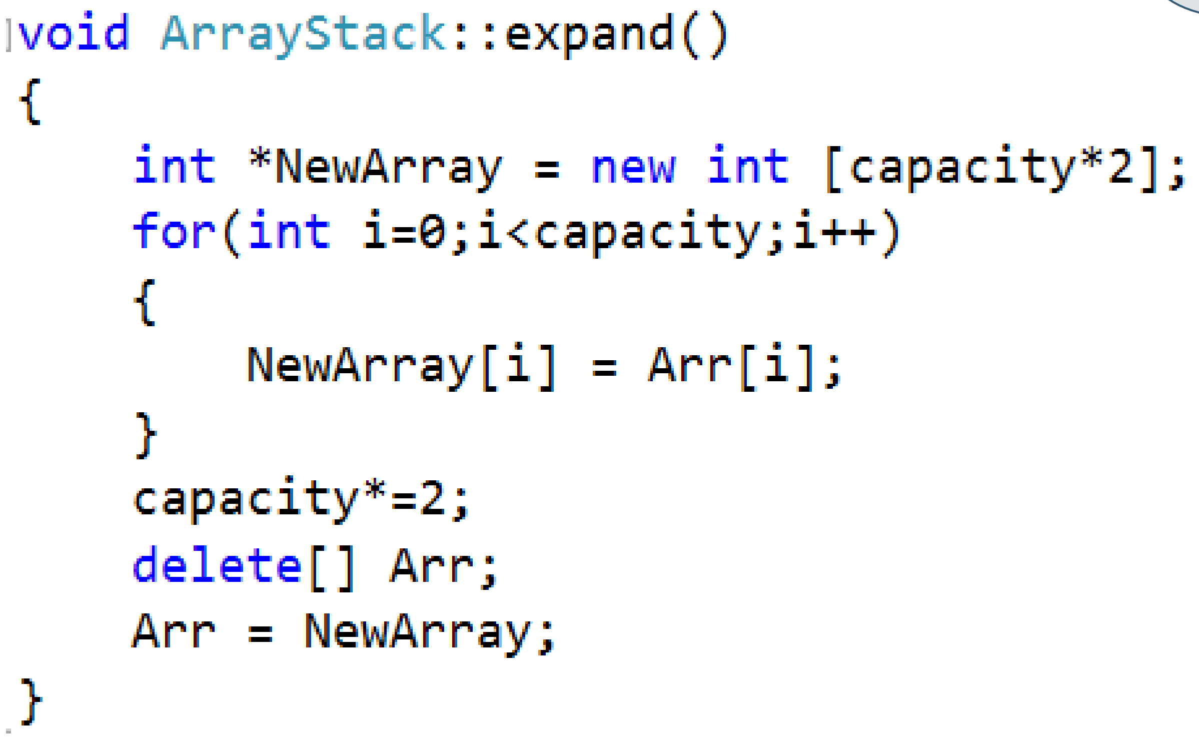
```
int ArrayStack::pop()
{
	Elements--;
	int Temp = Arr[Elements];
	return Temp;
}
```

TASK 1: PUSH , POP & EXPAND FUNCTION *“SOLUTION”*

15 minutes

ArrayStack class (.CppFile)

- **Expand**



```
void ArrayStack::expand()
{
    int *NewArray = new int [capacity*2];
    for(int i=0;i<capacity;i++)
    {
        NewArray[i] = Arr[i];
    }
    capacity*=2;
    delete[] Arr;
    Arr = NewArray;
}
```

TASK 2: DECIMAL TO BINARY CONVERTER

Implement a function that takes input parameters (decimal number) and works on it using **Stack** to convert it into a binary number.

15 minutes

TASK 2: DECIMAL TO BINARY CONVERTER *“SOLUTION”*

```
void convertDecimalToBinary(int decimalNumber) {
 ArrayStack stack;
 while (decimalNumber > 0) {
      stack.Push(decimalNumber%2);
      decimalNumber = decimalNumber/2;
 }
 while (!stack.isEmpty()) {
      cout<<stack.pop();
 }
 cout<<endl;
}
```

TASK 2: DECIMAL TO BINARY CONVERTER *“SOLUTION”*

Main.Cpp

```
int decimalNumber;

cout<<"Enter decimal Number to convert"<<endl;

cin>>decimalNumber;

convertDecimalToBinary(decimalNumber);
```

thank
you