```
#include <iostream>
using namespace std;
class Stack
{
       int* Stk;
       int Tos;
       int Size;
public :
    Stack ()
     {cout<<"\nparameter constructor"<<endl;
       Size = 5 ;
       Stk = new int [Size];
       Tos = 0 ;
     }
    Stack (int L)
     {cout<<"\nparameter constructor with one value "<<endl;
       Size = L ;
       Stk = new int [Size];
       Tos = 0 ;
     }
    ~Stack ()
     {cout<<"\nparameter destructor"<<endl;
       delete []Stk ;
     }
        bool isFull()
           { return (Tos ==5);}
        bool isEmpty()
            {return (Tos ==0);}

        void push(int n )
         {if (isFull() == false)
             {
                 Stk[Tos++] = n ;

             }
          else
            cout<<"stack is full "<<endl;
         }
       int pop( )
        {if (!isEmpty() )
         return Stk[--Tos]  ;
         else   {
          cout<<"stack is Empty "<<endl;
            return -1 ;
            }
        }
    void PrintStack()
    { int i;
      for (i=0;i<Tos ;i++)
      {
          cout<<Stk[i]<<endl;
      }
    }
    Stack Reverse()
    {
      cout <<"\n reversing ....";
      Stack stkRev ( this->Size);
      stkRev.Tos =this->Tos;
      for (int i=0; i<stkRev.Tos;i++)
         stkRev.Stk[i] =Stk[Tos-i-1];

      return stkRev;
    }
};
main()
{
   Stack S1;
   S1.push(10);
   S1.push(20);
   S1.push(30);
   S1.PrintStack();

   cout<<"\n____________________"<<endl;

   Stack ss= S1.Reverse();
   cout <<"\n NUM = "<<ss.pop();
   cout <<"\n NUM = "<<ss.pop();
   return 0;
}
```