A stack should have methods Push and Pop. This is just a small start to illustrate the concept. In this there is no stack, just nodes. We implement the idea of pushing in main:

```
#include <iostream>
#include <string>
using namespace std;
class IntNode {
   public:
     int data;               // will store information
     IntNode *next;
     IntNode(); //constructor
};
IntNode::IntNode() {
     next=NULL;
} //constructor

void main() {
  IntNode *top = NULL;
  //make a list of 3 integers
  int num,i;
  for(i=0;i<3;i++) {
    cout<<"Enter an integer: ";
    cin>>num;
    IntNode *temp;  //create a temporary node
    temp = (IntNode*)malloc(sizeof(IntNode)); //allocate space for node
    temp->data = num;             // store data(first field)
    temp->next=top;  // store the address of the pointer head(second field)
    top = temp;                  // transfer the address of 'temp' to 'top'
  } //3 numbers
  //look at the list
  IntNode *temp = top;
  while( temp!=NULL ) {
      cout<< temp->data<<" ";// show the data in the stack (linked list)
      temp = temp->next;   // tranfer the address of 'temp->next' to 'temp'
  } //loop until the end, that is when next is null
  system("pause");
}//main
```

If you run this and enter the numbers 8, 4, 3 it will print out 3, 4, 8. Each number is inserted at the front of the list. When we print the list we start at the front and work our way to the end.

In step 2 we have taken the code to add to the front of the list and created a method for the node called push(); We now use that in main and the result is the same.

```
#include <iostream>
#include <string>
using namespace std;
class IntNode {
   public:
     int data;               // will store information
     IntNode *next;
     IntNode(); //constructor
     IntNode* push(int num); //add to ftop of list
};
IntNode::IntNode() {
   next=NULL;
} //constructor
IntNode* IntNode::push(int num) {
   IntNode *cur;
   cur = (IntNode*)malloc(sizeof(IntNode)); //allocate space for node
   cur->data = num;             // store data
   cur->next=this;  //next is the current top of the list
   return cur;
} //push

void main() {
   IntNode *top = NULL;
   //make a list of 3 integers
   int num,i;
   for(i=0;i<3;i++) {
     cout<<"Enter an integer: ";
     cin>>num;
     top=top->push(num);
   } //3 numbers
  //look at the list
  IntNode *temp = top;
  while( temp!=NULL ) {
     cout<< temp->data<<" ";// show the data in the linked list
     temp = temp->next;   // transfer the address of 'temp->next' to 'temp'
  } //loop until the end, that is when next is null
  system("pause");
}//main
```