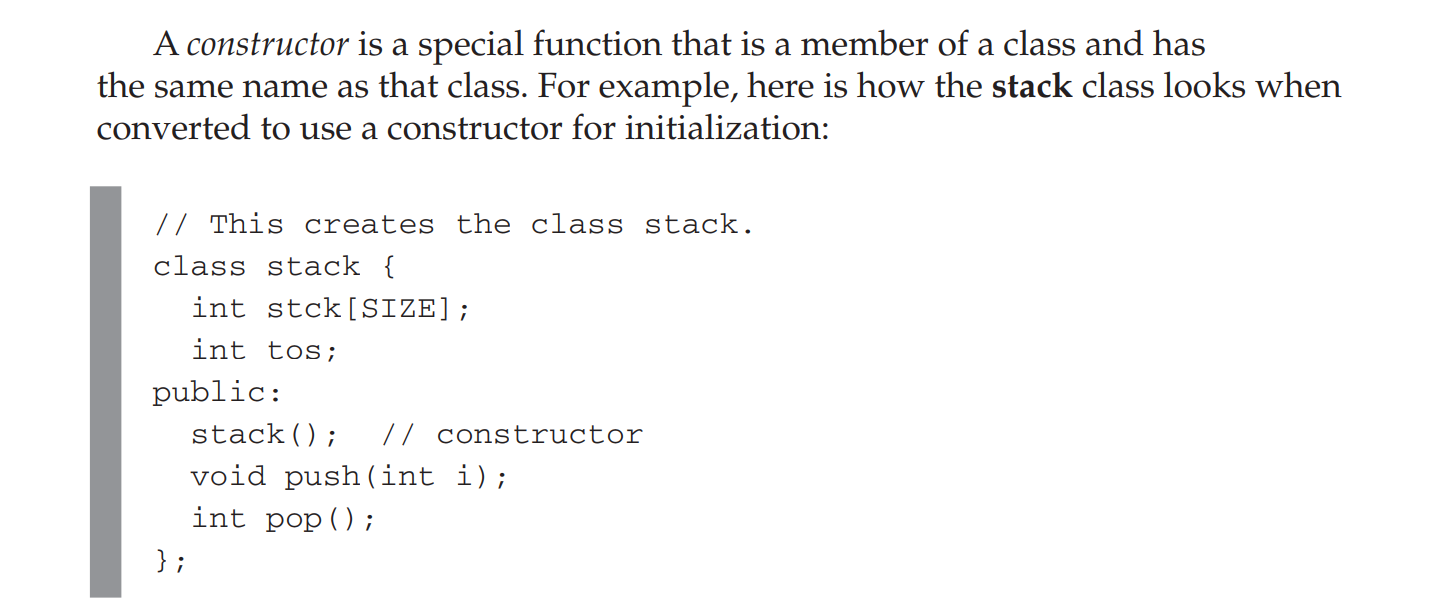
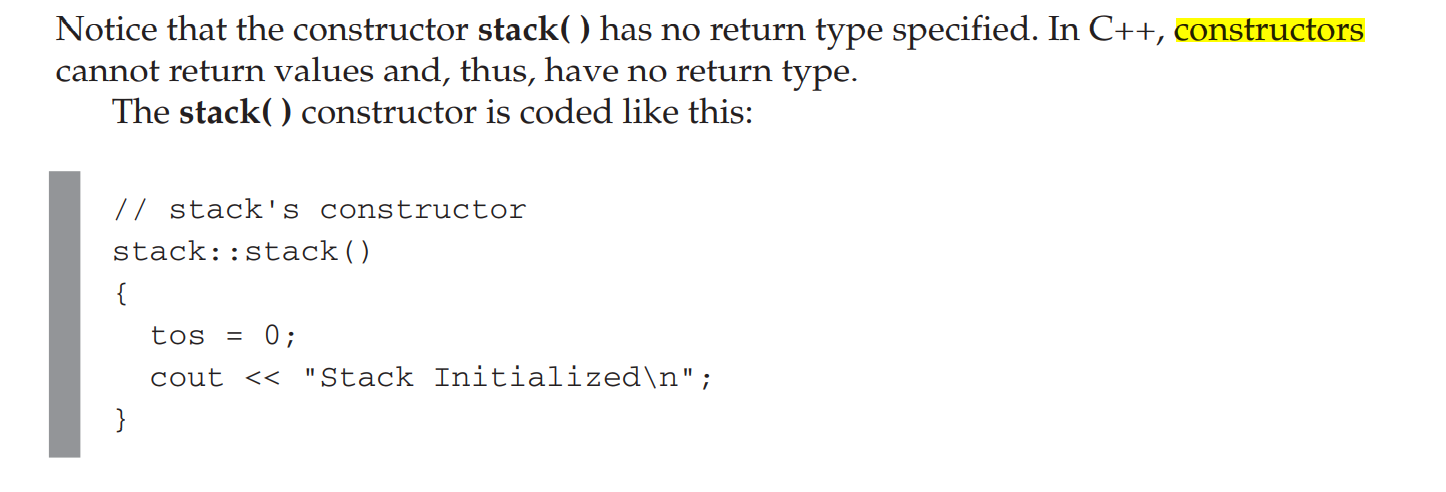
Constructors and Destructors

It is very common for some part of an object to require initialization before it can be used. For example, think back to the stack class developed earlier. Before the stack could be used, tos had to be set to zero. This was performed by using the function init( ). Because the requirement for initialization is so common, C++ allows objects to initialize themselves when they are created. This automatic initialization is performed through the use of a constructor function.





// Using a constructor and destructor.

#include <iostream>

using namespace std;

#define SIZE 100

// This creates the class stack.

class stack {

int stck[SIZE];

int tos;

public:stack(); // constructor

~stack(); // destructor

void push(int i);

int pop();

};

// stack's constructor

stack::stack()

{

tos = 0;

cout << "Stack Initialized\n";

}

// stack's destructor

stack::~stack()

{

cout << "Stack Destroyed\n";

}

void stack::push(int i)

{

if(tos==SIZE) {

cout << "Stack is full.\n";

return;

}

stck[tos] = i;

tos++;

}

int stack::pop()

{

if(tos==0) {

cout << "Stack underflow.\n";

return 0;

}

tos--;

return stck[tos];

}

int main()

{

stack a, b; // create two stack objects

a.push(1);

b.push(2);

a.push(3);

b.push(4);

cout << a.pop() << " ";

cout << a.pop() << " ";

cout << b.pop() << " ";

cout << b.pop() << "\n";

return 0;

}