Example Code of calling C++ functions from Assembly

Example:

```
.686              ;Target processor.  Use instructions for Pentium class machines

.MODEL FLAT, C    ;Use the flat memory model. Use C calling conventions
;INCLUDE Irvine32.inc

.STACK 2048       ;Define a stack segment of 1KB (Not required for this example)

askForInteger PROTO C
showInt PROTO C, value:SDWORD, outWidth:DWORD

OUT_WIDTH = 8
ENDING_POWER = 10
.data
intVal DWORD ?

.CODE             ;Indicates the start of a code segment.

clear PROC
xor eax, eax
xor ebx, ebx
INVOKE askForInteger ; call C++ function
mov intVal, eax ; save the integer

mov ecx, ENDING_POWER ; loop counter
L1:
push ecx ; save loop counter
shl intVal,1 ; multiply by 2
INVOKE showInt, intVal,OUT_WIDTH
pop ecx ; restore loop counter
loop L1
ret

clear ENDP
END
```

Cpp

```
#include "stdio.h"
#include <iostream>
#include <iomanip>
using namespace std;
// extern "C" instruct the compiler to use C calling conventions
extern"C" {
	void clear();
	int askForInteger();
	void showInt(int value, int width);
}

int main()
{
	clear( );
```

```
	return 0;
}

int askForInteger()
{
int n;
cout << "Enter an integer between 1 and 90,000:";
cin >> n;
return n;
}

// Display a signed integer with a specified width.
void showInt( int value, int width )
{
cout << width
cout << value;
}
```