Re-Distributing and Linking Object (.obj) Files (Functions; Subroutines) with C++ Mainline (Oct. 2002)

Consider the following C++ console application, called template.cpp.

```
#include <iostream>
using namespace std;

void stiffu(double &v1, double ar2[2][2]);
void stiffp(double &v1, double ar2[2][2]);

void main()
{
 double v1;
 v1 = 1.0;
 double ar2[2][2] = {1.0, 1.0, 1.0, 1.0};
 stiffu(v1, ar2);
 stiffp(v1, ar2);
 cout << ar2[1][1] << " " << v1 << endl;
}

void stiffu(double &v1, double ar2[2][2])
{
 ar2[1][1] = ar2[1][1] + 1.0;
 v1 = v1 + 1;
}
```

By accessing an externally compiled stiffp.obj file, the developer can execute the above program.

1. Select File – New – Project (Win32 project called test1).
2. Select console application (empty project).
3. Select View – Solution Explorer.
4. Right click on Source Files and select Add – Add New Item (.cpp file called test1).
5. Copy the above template.cpp into test1.cpp and save the file.
6. Another function, stiffp, has been compiled separately into object code, called stiffp.obj. It contains the same calling sequence and format of stiffu, as shown in the above template.cpp.
7. Copy the stiffp.obj file into the same test1 sub-directory, where the test1.cpp file is located.
8. Right Click on the test1 project in the Solution Explorer window.
9. Select Linker, Input and Additional Dependencies.
10. Enter the file name and directory of stiffp.obj in the Additional Dependencies.
11. Return back to test1.cpp, select Build Solution and Debug – Start to execute the full program.
12. This compilation / linking will use the external .obj file when calling stiffp from test1.cpp.