

```

1 #include <stdio.h>
2 #include <jni.h>
3
4 JNIEXPORT void JNICALL Java_CMyJava_Display(JNIEnv *env, jobject obj)
5 {
6     printf("\nI am in C\n");
7 }
8
9 JNIEXPORT void JNICALL Java_CMyJava_Add(JNIEnv *env, jobject obj, jint jiNo1, jint jiNo2)
10 {
11     int iAns = jiNo1 + jiNo2;
12
13     printf("\nAddition is %d\n", iAns);
14 }
15
16 JNIEXPORT void JNICALL Java_CMyJava_DisplayString
17     (JNIEnv *env, jobject obj, jstring js)
18 {
19     const char *cpStr = (*env)->GetStringUTFChars(env, js, NULL);
20     printf("\n%s\n", cpStr);
21 }
22
23 JNIEXPORT void JNICALL Java_CMyJava_CallMe
24     (JNIEnv *env, jobject obj)
25 {
26     jmethodID mid;
27
28     jclass cls = (*env)->FindClass(env, "CMyJava");
29     if(NULL == cls)
30         printf("\nfinding cls FAILED\n");
31     else
32         printf("\ncls FOUND\n");
33
34     mid = (*env)->GetStaticMethodID(env, cls, "DisplayJava", "()V");
35     if(mid != 0)
36         (*env)->CallStaticVoidMethod(env, cls, mid);
37     else
38         printf("\nMethod NOT FOUND\n");
39 }
40
41 JNIEXPORT void JNICALL Java_CMyJava_CallMeString
42     (JNIEnv *env, jobject obj)
43 {
44     jstring jstr;
45     jmethodID mid;
46
47     jclass cls = (*env)->FindClass(env, "CMyJava");
48     if(NULL == cls)
49         printf("\nfinding cls FAILED\n");
50     else
51         printf("\ncls FOUND\n");
52
53     jstr = (*env)->NewStringUTF(env, "Hello from C");
54
55     mid = (*env)->GetStaticMethodID(env, cls, "DisplayJavaString", "(Ljava/lang/String;)V");
56     if(mid != 0)
57         (*env)->CallStaticVoidMethod(env, cls, mid, jstr);
58     else
59         printf("\nMethod NOT FOUND\n");
60 }
61

```