Java Class (MathLibrary.java)

```
public class MathLibrary {

    // Declare native methods
    public native int add(int a, int b);
    public native int subtract(int a, int b);
    public native int multiply(int a, int b);
    public native int divide(int a, int b);

    static {
        // Load the shared library (libMathLibrary.so)
        System.loadLibrary("MathLibrary");
    }

    public static void main(String[] args) {
        MathLibrary math = new MathLibrary();

        int a = 20, b = 10;

        System.out.println("Addition: " + math.add(a, b));
        System.out.println("Subtraction: " + math.subtract(a, b));
        System.out.println("Multiplication: " + math.multiply(a, b));
        System.out.println("Division: " + math.divide(a, b));
    }
}
```

Generate JNI Header

```
# Compile Java class
javac MathLibrary.java

# Generate header file
javac -h . MathLibrary.java
```

Implement Native Methods in C (MathLibrary.c)

```
#include <jni.h>
#include "MathLibrary.h"
#include <stdio.h>

// Addition
JNIEXPORT jint JNICALL Java_MathLibrary_add(JNIEnv *env, jobject obj, jint a, jint b) {
    return a + b;
}
```

```
// Subtraction
JNIEXPORT jint JNICALL Java_MathLibrary_subtract(JNIEnv *env, jobject obj, jint a, jint b)
{
   return a - b;
}

// Multiplication
JNIEXPORT jint JNICALL Java_MathLibrary_multiply(JNIEnv *env, jobject obj, jint a, jint b)
{
   return a * b;
}

// Division
JNIEXPORT jint JNICALL Java_MathLibrary_divide(JNIEnv *env, jobject obj, jint a, jint b) {
   if (b == 0) return 0; // simple check to avoid divide by zero
   return a / b;
}
```

Compile Shared Library (.so) on Linux

```
gcc -fPIC -I${JAVA_HOME}/include -I${JAVA_HOME}/include/linux -shared -o
libMathLibrary.so MathLibrary.c
```

Run Java Program

```
# Make sure the shared library is found
export LD_LIBRARY_PATH=.:$LD_LIBRARY_PATH

# Run the program
java MathLibrary
```

Output:

```
Addition: 30
Subtraction: 10
Multiplication: 200
Division: 2
```