In the JNI framework, native functions are implemented in separate **.c or .cpp files**. When the JVM invokes the function, it passes a **JNIEnv pointer, a jclass pointer**, and any Java arguments declared by the Java method.

The *env* pointer is a structure that contains the interface to the JVM. It includes all of the functions necessary to interact with the JVM and to work with Java objects. Example JNI functions are converting native arrays to/from Java arrays, converting native strings to/from Java strings, instantiating objects, throwing exceptions, etc. Basically, anything that Java code can do can be done using JNIEnv.

JNIEXPORT and JNICALL are macros used to specify the calling and linkage convention of both JNI functions and native method implementations. The programmer must place the JNIEXPORT macro before the function return type and the JNICALL macro between the function name and the return type.