As far as theory is concerned you need to study:

1. Fast Fourier Transform
2. MFCC (Mel Frequency cepstrum coefficient)
3. DTW - Dynamic Time Warping / MSE - Mean Square Error

For implementing this step by step read this article for nokia mobile development, you can get good understanding of the flow and theory:http://www.developer.nokia.com/Community/Wiki/Sound_pattern_matching_using_Fast_Fourier_Transform_in_Windows_Phone

And for sample project you can find exactly the thing you are looking for:https://github.com/hfink/matchbox

Have a server doing audio fingerprinting computation that is not suitable for mobile device anyway. And then your mobile app uploads your files to the server and gets the analysis result for display. So I don't think programming language implementing it matters much. Following are a few AF implementations.

Java: http://www.redcode.nl/blog/2010/06/creating-shazam-in-java/

VC++: http://code.google.com/p/musicip-libofa/

c#: http://www.codeproject.com/KB/WPF/duplicates.aspx

Sounds like Shazam might help.

There are a few articles on the web talking about this, including this one that I read a while ago.

Using the Shazam technique for analysis would provide a great starting point for side-by-side comparison.

https://github.com/alexbw/iPhoneFFT

https://github.com/krafter/DetectingAudioFrequency

http://www.shazam.com/apps

http://en.wikipedia.org/wiki/Acoustic_fingerprint

http://www.codeproject.com/Articles/206507/Duplicates-detector-via-audio-fingerprinting http://www.addictivetips.com/ios/record-audio-from-5-minutes-in-the-past-heard-for-iphone/

http://www.w140.com/audio/