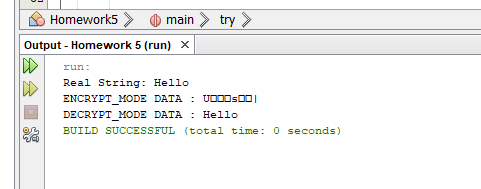
Use of a Broken or Risky Cryptographic Algorithm

It is important to periodically ensure that you aren’t using obsolete cryptography in your code. Some older encryption algorithms, once thought to require a billion years of computing time, can now be broken in days or hours. This include MD5, SHA1, DES, and many others that were once regarded as strong. For this example, we will take a look at DES.

DES (Data Encryption Standard) was highly influential in the advancement of modern cryptology. It is now considered to be obsolete mainly in part to its 56-bit key size.

Vulnerable-

For this demonstration, we can see that our application works the way it is supposed to. So, what is the problem? The problem is that it the algorithm has been broken and is no longer considered to be secure.



Mitigated-

In order to mitigate this vulnerability, we will select an algorithm that is currently considered to be strong by experts in the field. AES (Advanced Encryption Standard) is used by the US government to secure sensitive data. It is considered among the top ciphers. It is considered to be a lot more secure due to the 128-bit key size. With that being said, we will use it for our example. We will see how AES encryption and decryption can be done in Java.

