[0740] Finally!

Remember that exceptions are essentially invisible GOTO statements that can cause your program to leapfrog over lots of code on its way to the catch block. Usually that is fine, however, sometimes there are times where you need some special code to always, always execute. For example, you might need to always, always close a database connection - even if there was an exception.

In cases like that, you can add a "finally" clause after the "catch" clause. Any code in the finally clause is then guaranteed to run regardless of whether there was an exception or not.

For example:

```
try
{
   DBConnection conn = new DBConnection("MyDatabaseOfDoom");

   MethodThatCausesAnException();

} catch (MethodsException e)
{

} finally
{
  conn.Close();
}
```

Just remember that conn.Close() will be called regardless of whether there is an exception or not.