```
//File Shredder Code Sample (C# 4.0)

//It has been a while since I last posted a code sample.
//So here is one for you guys to take a look at.
//Below is the file/directory shredder class.

class FileUtilities
{
   private static Random Randomizer = new Random();

   public static bool Shred(string directoryPath, bool shouldDelete = true)
   {
      bool success = true;
      try
      {
         string[] files = Directory.GetFiles(directoryPath, "*", SearchOption.AllDirectories);
         foreach (string file in files)
         {
            success &= FileUtilities.ShredFile(file, shouldDelete);
         }

         string[] directories = Directory.GetDirectories(directoryPath,
                                          "*",
                                          SearchOption.AllDirectories).OrderByDescending(str =>
                                                                  str.Split('\\').Length - 1).ToArray();
         foreach (string directory in directories)
         {
            success &= FileUtilities.ShredDirectory(directory, shouldDelete);
```

```
        }

        success &= FileUtilities.ShredDirectory(directoryPath, shouldDelete);
      }
      catch
      {
        success = false;
      }
      return success;
    }

    public static bool ShredDirectory(string directoryPath, bool shouldDelete = true)
    {
      bool success = true;
      try
      {
        DirectoryInfo directoryInfo = new DirectoryInfo(directoryPath);

        string[] directoryBits = directoryPath.Split('\\');
        directoryBits[directoryBits.Length - 1] =
FileUtilities.RandomName(directoryInfo.Name.Length);

        string newDirectoryPath = String.Join("\\", directoryBits);
        directoryInfo.MoveTo(newDirectoryPath);

        if (shouldDelete)
        {
          directoryInfo.Delete();
```

```
        }
      }
      catch
      {
        success = false;
      }
      return success;
    }

    public static bool ShredFile(string filePath, bool shouldDelete = true)
    {
      bool success = true;
      try
      {
        FileStream fs = new FileStream(filePath, FileMode.Open, FileAccess.Write);
        for (long i = 0; i < fs.Length; i++)
        {
          fs.WriteByte((byte)Randomizer.Next(0, 255));
        }
        fs.Close();

        FileInfo fileInfo = new FileInfo(filePath);
        fileInfo.MoveTo(fileInfo.DirectoryName + @"\" +
FileUtilities.RandomName(fileInfo.Name.Length));

        if (shouldDelete)
        {
          fileInfo.Delete();
```

```
        }
      }
      catch
      {
        success = false;
      }
      return success;
    }

    private static string RandomName(int length)
    {
      string fileNameChars =
"abcdefghijklmnopqrstuvwxyz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ~!@¤#$£§%^&()+`={}][
;,";
      string newFileName = String.Empty;
      for (int i = 0; i < length; i++)
      {
        newFileName += fileNameChars[Randomizer.Next(0, fileNameChars.Length)];
      }
      return newFileName;
    }
}
```

//The usage calls for these functions as as follows:

bool success = FileUtilities.Shred(@"D:\Temp"); // This method will recursively shred a file / folder.

bool success = FileUtilities.ShredDirectory(@"D:\Temp"); // This method will shred the contents of a folder and the root folder.

```
bool success = FileUtilities.ShredFile(@"D:\Temp\test.txt"); // This method will shred a specified file.
```