
General microbiology by ystanier 5th edition pdf 35

[Download](#)

A: Because.txt files are plain text, so you can open them in a plain text editor such as notepad to see their content and then remove them. For more in depth instructions visit [Remove all.txt files](#) A: You can use Powershell if you have installed it on your Windows system. For example: `$files = dir foreach ($file in $files){ if ($file.name -match '.txt'){ Remove-Item -Path $file.FullName -Force } }` Q: Is it possible to create objects at runtime in .Net? I have multiple classes that share a method in which I do some data operations. I would like to factor out this shared code so that the classes will be able to use an interface that exposes the shared method. But I am not sure if it's a good approach because I'd like to keep the static methods out of the classes where the objects are created. Here is a concrete example: `public interface IOrderDataProcessor { OrderStart CreateOrderObject(string myOrder); } public class AOrderDataProcessor : IOrderDataProcessor { public OrderStart CreateOrderObject(string myOrder) { //do some data operations to create an order } } public class BOrderDataProcessor : IOrderDataProcessor { public OrderStart CreateOrderObject(string myOrder) { //do some data operations to create an order } }` Let's assume that at runtime I want to create two objects of class AOrderDataProcessor and one object of class BOrderDataProcessor. Is it possible to create objects using runtime type of some variable? A: Use the Activator class. Here's a simple example: `var dataProcessor = (IOrderDataProcessor)Activator.CreateInstance(typeof(AOrderDataProcessor));` A: No, you cannot create objects at runtime. A good approach would be a Factory or a Factory method as an extension to your interface, or something along the lines of this: `public interface IOrderDataProcessor {`

