Module Review and Takeaways

In this module, you became familiar with various techniques that you can use to eliminate bugs from your ASP.NET MVC 5 web application. This begins early in the development phase of the project, when you define unit tests that ensure that each method and class in your application behaves precisely as designed. Unit tests are automatically rerun as you build the application so you can be sure that methods and classes that did work are not broken by later coding errors. Exceptions arise in even the most well-tested applications because of unforeseen circumstances. You also saw how to ensure that exceptions are handled smoothly, and how error logs are optionally stored for later analysis.

Best Practice: If you are using TDD or Extreme Programming, define each test before you write the code that implements a requirement. Use the test as a full specification that your code must satisfy. This requires a full understanding of the design.

Best Practice: Investigate and choose a mocking framework to help you create test double objects for use in unit tests. Though it may take time to select the best framework and to learn how to code mock objects, your time investment will be worth it over the life of the project.

Best Practice: Do not be tempted to skip unit tests when under time pressure. Doing so can introduce bugs and errors into your system and result in more time being spent debugging.

Common Issues and Troubleshooting Tips

Common Issue	Troubleshooting Tip
No information appears in the IntelliTrace window.	

Review Question(s)

Question: You want to ensure that the PhotoController object passes a single Photo object to the Display view, when a user calls the **Search** action for an existing photo title. What unit tests should you create to check this functionality?

Tools

NinJect, StructureMap. These are Inversion of Control (IoC) containers, also known as Dependency Injection (DI) frameworks. They create non-test implementations of interfaces in your web application.

Moq, RhinoMocks, NSubstitute. These are mocking frameworks. They automate the creation of test doubles for unit tests.

IntelliTrace. This is a part of Visual Studio that displays application state at the point of an exception or break, and at earlier times.

Health Monitoring. This part of ASP.NET can store health events in a database, log, or other locations for later analysis.

ELMAH. This exception logging tool can store exceptions in database tables, XML files, and elsewhere, and enable administrators to view exception details on a webpage.