**Code Analysis & Unit Testing**

In this course module, you will explore the following topics:

1. Code Analysis
2. Debugging
3. Unit Testing

**Content**

* **Code Analysis**

Please refer to the following presentations from the “C\_Coding” compressed file. While studying you can follow the order specified here.

1. ACP\_Coding\_Foundation\_Rel\_1.0
2. ACP\_Coding\_Proficient\_Rel\_1.0
3. ACP\_Coding\_Advanced\_Rel\_1.0

Please refer to the following demo-documents along with the sample codes in the “code\_analysis” compressed file to understand the tools used for code analysis and debugging.

The tools covered are

1. Static code analysis:Klocwork and Splint
2. Debugging:gdb
3. Memory Leak : Valgrind

* **Unit Testing and Code coverage**
  1. To understand the concept of Test Driven Development, refer to the following Lynda course: [**Foundations of Programming: Test-Driven Development**](https://www.lynda.com/Developer-Programming-Foundations-tutorials/Foundations-Programming-Test-Driven-Development/124398-2.html)
  2. Please refer to the following presentation for Unit Testing and code coverage in the Unit\_Testing compressed file

1. UT and Code Coverage in C\_4.0

Refer to the following demo-documents along with sample codes in the Unit\_Testing compressed file.

The tools covered are

1. Code Coverage:gcov
2. CUnit

**Assignments**

In continuation of the case study shared along with Design Analysis and Algorithms, a part of code is available in the compressed file

Refer to the code and perform the following:

1. Analyze the code without executing for any possible bugs and correct the code.
2. Fix the code as per the Aricent code review checklist available on the Aricent intranet(QMS) and find the code review readiness .
3. Find out about the code health and Coding guidelines issues and correct them.
4. Debug the code to find out about the possible bugs and make it a running application.
5. Write Unit Test cases using CUnit framework.

**Note:** The code uses Glib which is an open source data structure library for which the sample code is available inside the compressed “Sample\_glibc” directory. You can refer to the below url: (<http://www.ibm.com/developerworks/linux/tutorials/l-glib/>)

You should complete the assignments offline and need not submit the assignments.