

Course Name	Unit Testing Using Unity Framework
Required Software	Install below software before the session: <ol style="list-style-type: none"> 1. VS code 2. Windows OS + wsl2 (or) Linux OS
Syllabus	<ol style="list-style-type: none"> 1. Introduction to Unit Testing <ol style="list-style-type: none"> a. What is Unit Testing? b. Benefits of Unit Testing in Embedded and Systems Programming c. Overview of Test-Driven Development (TDD) d. Challenges of Unit Testing in C 2. Multifile and Makefile <ol style="list-style-type: none"> a. Creating and using header/source files b. Compilation and linking process c. Writing and using Makefiles d. Targets, dependencies, commands e. Variables and pattern rules f. Integrating Unity tests in multifile projects using Makefiles 3. Getting Started with Unity Framework <ol style="list-style-type: none"> a. Introduction to Unity Framework b. Installing Unity c. Folder structure and test organization 4. Writing Test Cases <ol style="list-style-type: none"> a. Unity Macros: TEST_ASSERT_EQUAL, TEST_ASSERT_TRUE, etc. b. Testing functions with: <ol style="list-style-type: none"> I. Return values II. Pointers III. Structures IV. Arrays c. Grouping and organizing tests 5. Test Fixtures and Setup <ol style="list-style-type: none"> a. Using setUp() and tearDown() functions b. Managing test dependencies c. Isolating test environments 6. Automating Tests <ol style="list-style-type: none"> a. Running tests via command line b. Integrating with build systems (Makefile)
Mandatory Assignment	<ol style="list-style-type: none"> 1. Daily activities shared after the session needs to be submitted within timelines. 2. Write unit test cases using unity for all the C programming activities from the previous module.
End Module Assessment	MCQ based Hacker earth assessment with Camera enabled.
Additional Learning & Practice Resources	Embetroneix/Unit-Testing