| | |
|---|---|
| Course Code | CS 4XX/6XX |
| Title of the Course | Software Correctness and Certification |
| Course Category | Department Elective |
| Credit Structure | L-T-P–Credits<br>2-1-0-3 |
| Department | Computer Science and Engineering |
| Pre-requisite | Automata Theory and Logic CS 202, Computer Programming CS 101 |
| Scope | 1. To understand the significance of correctness, safety, and security in open-source software.<br>2. To learn techniques of formal verification and static code analysis. |
| Course Syllabus | • Foundations of Software Correctness: Fundamentals of software correctness-safety-liveness, formal software verification, verification techniques: deductive vs algorithmic and hybrid approaches<br>• Formal Verification Techniques: Modelchecking and Theorem proving, Program analysis: Static vs dynamic, Case studies, synergistic verification and validation<br>• Verification- Languages, Tools, and Standards: property specification languages, system modelling langauges, and tools, domain specific Coding standards and implications<br>• Collaborative Verification and Licensing Considerations: openness vs verifiability, Industrial case studies, Application in responsibility specification and analysis<br>• Correctness-Driven Software Lifecycle: Integration of correctness in the software development lifecycle\, correctness-certification of open-source projects |
| Suggested Books | Textbooks:<br>1. Edmund M. Clarke Jr., Orna Grumberg, Daniel Kroening, Doron Peled and Helmut Veith, “**Model Checking”,** the MIT Press, 2018, ISBN: 9780262038836<br>2. Xavier Rival, Kwangkeun Yi, “**Introduction to Static Analysis: An Abstract Interpretation Perspective”,** The MIT Press, 2020, ISBN:9780262043410<br>Reference books:<br>3. Steven Weber, “**The Success of Open Source”**, Harvard University Press, USA, 2005, ISBN: 9780674018587 |