**Detailed Syllabus**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Code** | **15B17CI573** | **Semester** Odd | | **Semester** 5 **Session** 2019 -2020 Month from July’19 to December’19 | |
| **Course Name** | **Software Engineering Lab** | | | | |
| **Credits** | 1 | | **Contact Hours** | | 2 |

|  |  |  |
| --- | --- | --- |
| **Faculty (Names)** | **Coordinator(s)** | Dr. Shruti Jaiswal, Sarishty Gupta |
| **Teacher(s) (Alphabetically)** | Amarjeet Prajapati, Anuja Arora, Himanshu Aggarwal, Nitin Shukla, Sarishty Gupta, Shruti Jaiswal |

|  |  |  |
| --- | --- | --- |
| **COURSE OUTCOMES** | | **COGNITIVE LEVELS** |
| **C371.1** | Explain software engineering principles and software process models for project development, software requirements specification for a software project | Understand Level (Level II) |
| **C371.2** | Apply Software Design and modeling. | Apply Level (Level III) |
| **C371.3** | Apply Software Optimizing and Refactoring | Apply Level (Level III) |
| **C371.4** | Apply testing principles and implement various testing procedures | Apply Level (Level III) |
| **C371.5** | Creation of software using software engineering principals | Create (level VI) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Module No.** | **Title of the Module** | **List of Experiments** | **CO** |
| **1.** | Introduction to Software Engineering Principals | Introduction to software engineering Principles (evolution, failures, changing nature of software, software myths, product, process, software crisis and need of testing), Software process models (build and fix model, waterfall model, Incremental process model, Evolutionary- Prototype and Spiral models, Agile models – extreme programming and scrum, selection of a life cycle model), PSP, TSP. Types of requirement, Feasibility studies, Requirement Elicitation, Analysis, Specification, SRS, Requirement Verification and Validation. | 1 |
| **2.** | Software Design and modeling. | Use case diagram, State diagram, Activity Diagram, Class Diagram, Sequence diagram, Collaboration diagram, Deployment Diagram, Event trace diagram. Size oriented metrics, LOC, token count, Function Count, cost estimation, data structure metrics, Halstead’s Software Metric, Information Flow Metric, Overview of Quality Standards like ISO 9001, SEI-CMM, COCOMO, COCOMO-II, Software risk management | 2 |
| **3.** | Software Optimizing and Refactoring | Coding standards and guidelines, Code checklist, Code Refactoring and Code optimization | 3 |
| **4.** | Software Testing | Black box testing techniques: Equivalence class testing, Boundary value analysis, Decision table testing, Cause effect graphing, White box testing: Path testing, Data flow and mutation testing, Levels of testing- unit testing, integration and system testing, Debugging- techniques, approaches, tools & standards. | 4 |
|  |  |  |  |
| **Evaluation Criteria**  **Components Maximum Marks**  Lab Test 1 20  Lab Test 2 20  Day-to-Day(Evaluations, Viva, 60  Quiz, Attendance, Project)  **Total 100** | | | |
|  | | | |
|  | | | |

|  |  |
| --- | --- |
| **Recommended Reading material:** Author(s), Title, Edition, Publisher, Year of Publication etc. (Text books, Reference Books, Journals, Reports, Websites etc. in the IEEE format) | |
| **1.** | Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave Macmillan, 2005. |
| **2.** | Jalote, Pankaj. An integrated approach to software engineering. Springer Science & Business Media, 2012. |
| **3.** | KK Aggarwal, Software Engineering, 2001. |
| **4.** | David Solomon and Mark Russinovich ,” Inside Microsoft Windows 2000”, Third Edition, Micorosoft Press |
| **5.** | https://www.tutorialspoint.com/software\_engineering/ |
| **6.** | ACM/IEEE transactions on Software Engineering |
| **7.** | ACM Transactions on Software Engineering Methodology |
| **8.** | Springer Journal of Empirical Software Engineering |
| **9.** | Springer Journal of Software and Systems Modeling |