**Detailed Syllabus**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Code** | 15B11CI513 | **Semester** Odd  **(specify Odd/Even)** | | **Semester** 5th **Session** 2019 -2020 Month from July 19 to Dec 19 | |
| **Course Name** | **Software Engineering** | | | | |
| **Credits** | 4 (3-1-0) | | **Contact Hours** | | 3+1 |

|  |  |  |
| --- | --- | --- |
| **Faculty (Names)** | **Coordinator(s)** | Sangeeta (62), Himanshu Agrawal (128) |
| **Teacher(s) (Alphabetically)** | Anuja Arora, Sarishty Gupta (62)  Amritpal Singh, Nitin Shukla (128) |

|  |  |  |
| --- | --- | --- |
| **COURSE OUTCOMES** | | **COGNITIVE LEVELS** |
| **C311.1** | Explain software engineering principles and software process models for project development. | Remembering(Level 1) |
| **C311.2** | Identify functional and non-functional requirements of a software project and design document software requirements specification. | Understand (Level 2) |
| **C311.3** | Design, represent and document software requirements specification. Plan and execute activities for a software project. | Create (Level 6) |
| **C311.4** | Apply UML modeling for software design from software requirements specification. | Apply(Level 3) |
| **C311.5** | Analyze code checklist. Perform code Reviews, Code Refactoring, and Code optimization, design pattern | Analyze(Level 4) |
| **C311.6** | Apply testing principles, develop and implement various manual and automated testing procedures, formal methods | Apply(Level 3) |
| **C311.7** | Evaluate software in terms of general software quality attributes and possible trade-offs presented within the given problem. | Evaluate(Level 5) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Module No.** | **Title of the Module** | **Topics in the Module** | **No. of Lectures for the module** |
| **1.** | *Unit-1* | ***Introduction to Software Engineering:*** Introduction to software engineering Principles, Software process models(build and fix model,waterfall model, Incremental process model, Evolutionary- Prototype and Spiral models, Agile Models (tools study), PSP, TSP, Software Reengineering.Project planning, Project Scheduling: network diagram, Gant Chart, CPM and PERT. [7L] | 7 |
| **2.** | ***Unit-2*** | ***Requirement Engineering:***  Types of requirement, Requirement Elicitation, Analysis, Specification, SRS, Requirement Verification and Validation.**[4L]** | **4** |
| **3.** | ***Unit-3*** | ***Software Design:***  Use case diagram, State diagram, Activity Diagram, Class Diagram, Sequence diagram, Collaboration diagram, Deployment Diagram, Component Diagram and Package diagram. Design Modularity: Coupling Cohesion. **[7L]** | **5** |
| **4.** | ***Unit-4*** | ***Software Construction:***  Coding standards and guidelines, Code checklist, Code Reviews, Code Refactoring, Code optimization.Design pattern  Modern programming environments (Code search, Programming using library components and their APIs),  Program comprehension; Program correctness, Defensive programming. **[8L]** | **9** |
| **5.** | ***Unit-5*** | ***Software Metrics:***  Size-Oriented Metric, Function-oriented Metric, Halstead’s Software Metric, Information Flow Metric, Object-oriented Metric, Class-Oriented Metric, COCOMO Model. **[7L]** | **7** |
| **6.** | ***Unit-6*** | ***Software Testing:***  White-Box Testing, Basis Path Testing, Control Structure Testing: Condition Testing, Data Flow Testing, Loop Testing, Black-Box Testing: Equivalence class partitioning, Boundary Value Analysis, Decision table testing, Cause effect graphing, Mutation Testing and regression Testing, formal methods**[9L]** | **10** |
| **Total number of Lectures** | | | **42** |
| **Evaluation Criteria**  **Components Maximum Marks**  T1 20  T2 20  End Semester Examination 35  TA 25 (Assignemnts/Tutorial : 20  Attendance : 5)  **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) | |
| **Text Book(s):** | |
| **1.** | Roger S. Pressman, “Software Engineering: A practitioner approach”, Fifth Edition-TMH International . |
| **2.** | Sommerville , “Software Engineering” , Seventh Edition - Addison Wesley. |
| **Reference Book(s):** | |
| **3.** | [Grady Booch](http://dret.net/biblio/authors#GradyBooch), [James Rumbaugh](http://dret.net/biblio/authors#JamesRumbaugh), [Ivar Jacobson](http://dret.net/biblio/authors#IvarJacobson), [The Unified Modeling Language User Guide](http://dret.net/biblio/titles#boo05), Addison Wesley, Reading, Massachusetts, May 2005 |
| **4.** | Richard Thayer , “Software Engineering Project Management”, Second Edition -Wiley-IEEE Computer Society Press. |
| **5.** | B. Bezier, “Software Testing Techniques”, Second Edition- International Thomson Computer Press. |
| **6.** | Pankaj Jalote, “An Integrated Approach to Software Engineering” Third addition , Springer Press |