NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

<table>
<tr><td colspan="4" align="center">COURSE PLAN – PART I</td></tr>
<tr><td>Name of the programme and specialization</td><td colspan="3">B.Tech Computer Science and Engineering</td></tr>
<tr><td>Course Title</td><td colspan="3">Software Engineering</td></tr>
<tr><td>Course Code</td><td>CSMI15</td><td>No. of Credits</td><td align="center">3</td></tr>
<tr><td>Course Code of Pre-requisite subject(s)</td><td colspan="3">NIL</td></tr>
<tr><td>Session</td><td>January 2019</td><td>Section (if, applicable)</td><td align="center">A & B</td></tr>
<tr><td>Name of Faculty</td><td>Mrs. B. Preetha / Mr. Arumugam J</td><td>Department</td><td align="center">CSE</td></tr>
<tr><td>Email</td><td>preetha@nitt.edu
arumugamj@nitt.edu</td><td>Telephone No.</td><td align="center">-</td></tr>
<tr><td>Name of Course Coordinator(s) (if, applicable)</td><td colspan="3"></td></tr>
<tr><td>E-mail</td><td></td><td>Telephone No.</td><td></td></tr>
<tr><td>Course Type</td><td colspan="3">Elective course (Minor)</td></tr>
</table>

Syllabus (approved in Senate)

Unit - I

The Evolving role of Software, The changing Nature of Software, Legacy software, A generic view of process, A layered Technology, A Process Framework, Software Development Life cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models, Incremental Process Model, Concurrent Development Model, The Unified Process.

Unit-II

Software Requirements, Functional & non-functional, Software requirements document, Requirement engineering process: Feasibility studies, Elicitation, Validation & Management, Software prototyping, Analysis and modeling.

Unit-III

Design Concepts and Principles –Modular design –Design heuristic –S/W architecture –Data design – Architectural design –Transform & Transaction mapping–Introduction to SCM process –Software Configuration Items.

Unit –IV

Software Testing, Taxonomy of S/W testing, Black box testing, Testing boundary conditions, Structural testing, Regression testing, S/W testing strategies : Unit testing, Integration testing Validation testing, System testing and debugging.

Unit-V

Software Project Management, S/W cost estimation, Function point models, COCOMO model, Project Scheduling, S/W maintenance.

COURSE OBJECTIVES

➢ To understand the Software Engineering Practice& Process Models
➢ To understand Design Engineering, Web applications, and Software Project Management
➢ To gain knowledge of the overall project activities

COURSE OUTCOMES (CO)

➢ Ability to enhance the software project management skills
➢ Ability to comprehend the systematic methodologies involved in SE
➢ Ability to design and develop a software product in accordance with SE principles

Course Outcome (CO)	Aligned programme Outcome
Ability to enhance the software project management skills	1 , 2 , 3, 7
Ability to comprehend the systematic methodologies involved in SE	1, 2, 4, 5
Ability to design and develop a software product in accordance with SE principles	1, 2, 3, 5

COURSE PLAN – PART II

COURSE OVERVIEW

This course mainly describes about various software Engineering Practice & Process models for an efficient software design.

COURSE TEACHING AND LEARNING ACTIVITIES

S.No.	Week	Topic	Mode of Delivery
1.	I Week	The Evolving role of Software, The changing Nature of Software, Legacy software, A generic view of process: A layered Technology, A Process Framework. Software Development Life cycle (SDLC) Models: Water Fall Model.	Chalk and Talk
2.	II Week	Incremental Process Model: Iterative Enhancement Models. Evolutionary Process Models, Prototype Model, Spiral Model, Concurrent Development Model.	Chalk and Talk
3.	III Week	The Unified Process. Software Requirements, Functional & non-functional.	Chalk and Talk
4.	IV Week	Software requirements document, Requirement engineering process: Feasibility studies, Elicitation, Validation & Management.	Chalk and Talk
5.	V Week	Software prototyping, Analysis and modeling. Design Concepts and Principles, Modular design, Design heuristic.	Chalk and Talk
6.	VI Week	S/W architecture - Data design, Architectural design - Transform & Transaction mapping.	Chalk and Talk

7.	VII Week	Introduction to SCM process –Software Configuration Items.	Chalk and Talk
8.	VIII Week	Software Testing, Taxonomy of S/W testing, Black box testing, Testing boundary conditions.	Chalk and Talk
9.	IX Week	Structural testing, Regression testing, S/W testing strategies, Unit testing.	Chalk and Talk
10.	X Week	Integration testing Validation testing, System testing and debugging.	Chalk and Talk
11.	XI Week	Software Project Management, Function point models.	Chalk and Talk
12.	XII Week	S/W cost estimation, COCOMO model.	Chalk and Talk
13.	XIII Week	Project Scheduling, S/W maintenance.	Chalk and Talk

Text Books
1. Roger S.Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill International edition, Seventh edition, 2009.
2. Ian Sommerville, Software Engineering, 9th Edition, Pearson Education, 2011.

Reference books
1. Pankaj Jalote, "An Integrated Approach to software Engineering", Springer Verlag, 1997.
2. Pfleeger and Lawrence Software Engineering: Theory and Practice, Pearson Education, second edition, 2001
3. Rajib Mall, "Fundamentals of Software Engineering", PHI Publication, 3rdedition, 2009.

COURSE ASSESSMENT METHODS-THEORY (shall range from 4 to 6)

S.No.	Mode of Assessment	Week/Date	Duration	% Weightage
1.	Cycle Test-1	3rd week of Feb	1 hour	20%
2.	Cycle Test-2	1st week of April	1 hour	20%
3.	Assignment*	1st week of March	-	10%
CPA	Compensation Assessment*	4th week of April	1 hour	20%
4.	Final Assessment*	2nd week of May	3 hours	50%
TOTAL				**100%**

*mandatory

COURSE EXIT SURVEY (mention the ways in which the feedback about the course shall be assessed)

1. Students' feedback through class committee meetings.
2. Feedback questionnaire from students – from MIS at the end of the semester.

COURSE POLICY (preferred mode of correspondence with students, compensation assessment policy to be specified)

MODE OF CORRESPONDENCE (email/ phone etc)
Mode of Correspondence through Phone.

COMPENSATION ASSESSMENT POLICY
In case of emergency, the student should submit compensatory assignments on submission of appropriate documents as proof. Compensatory assessments would be framed according to the time frame available and the assessment task missed by the students.

ATTENDANCE POLICY (A uniform attendance policy as specified below shall be followed)

- **At least 75% attendance in each course is mandatory.**

- **A maximum of 10% shall be allowed under On Duty (OD) category.**

- Students with **less than 65% of attendance** shall be prevented from writing the final assessment and **shall be awarded 'V' grade.**

ACADEMIC DISHONESTY & PLAGIARISM

- Possessing a mobile phone, carrying bits of paper, talking to other students, copying from others during an assessment will be treated as punishable dishonesty.

- Zero mark to be awarded for the offenders. For copying from another student, both students get the same penalty of zero mark.

- The departmental disciplinary committee including the course faculty member, PAC chairperson and the HoD, as members shall verify the facts of the malpractice and award the punishment if the student is found guilty. The report shall be submitted to the Academic office.

- The above policy against academic dishonesty shall be applicable for all the programmers.

ADDITIONAL INFORMATION

The students can get their doubts clarified at any time with their faculty member.

FOR APPROVAL

Course Faculty _____ CC-Chairperson _____ HOD _____
22/1/19