

UKA TARSADIA UNIVERSITY  
B.C.A. (4<sup>th</sup> Semester) Syllabus, 2019-2020

Course Code: CS4007

Course Credits: 04

Course Title: Introduction to Software Engineering

[Lectures: 04, Tutorial: 00, Practical: 00]

Prerequisites: Object Oriented Programming and Information Systems concepts.

Prerequisites By Topics:

Objectives: To introduce the fundamentals of software engineering for efficient development of software using software process model and software testing strategies with a test automation tool.

**Course Outcomes: Upon completion of the course, students shall be able to**

CO1: Comprehend the importance and fundamentals of Software Engineering and requirement engineering.

CO2: Compare, contrast and analyze the conventional and object-oriented software process models with its applicability.

CO3: Recognize the prominence of Agile Process Models for software development.

CO4: Describe and apply the basic principles of software design for software design, refinement and implementation.

CO5: Discuss the needs, importance of system testing.

CO6: Analyze and relevant testing strategies through design and execution of test case by utilizing test automation tools.

**Course Objective and Course Outcomes Mapping:**

To introduce the fundamentals of software engineering for efficient development of software using software process model:

CO1, CO2 and CO3

Software testing strategies with test automation tool: CO4, CO5 and CO6

**Programme Outcomes and Course Outcomes mapping:**

Course Outcomes	Programme Outcomes				
	P01	P02	P03	P04	P05
CO1	✓	✓			✓
CO2			✓		
CO3			✓		
CO4			✓		
CO5			✓		
CO6			✓		

- |   |               |
|---|---------------|
| <b>1 Software Engineering Fundamentals</b><br>1.1. Concepts, Needs<br>1.2. Principals of Software engineering<br>1.3. Phases of Software Engineering: Requirement Analysis, Modelling, Design, Coding, Testing<br>Deployment and Maintenance<br>1.4. Requirement Analysis and Engineering: Overview   | <b>[10 %]</b> |
| <b>2 Process Modeling</b><br>2.1. Conceptual overview of Process, Process framework<br>2.2. Process Models: Introduction and Types<br>2.3. Conventional and Object Oriented Models : Differentiation, Classification<br>2.4. Agile Paradigm : Overview<br>2.5. Agile Process Models : Types   | <b>[18%]</b>  |
| <b>3 Software Design</b><br>3.1 Introduction and Overview<br>3.2 Quality Guidelines, Design and Principles<br>3.3 Attributes of Software Design: Functionality, Usability, Reliability, Performance, Supportability<br>3.4 Cohesion and Coupling : Types and Usage<br>3.5 Identification of Design Classes, Relationships, Construction and Refinement<br>3.6 Mapping Design Classes and Data Structure | <b>[22%]</b>  |
| <b>4 System Testing</b><br>4.1 Introduction, Needs<br>4.2 Testing versus Quality and Debugging<br>4.3 Testability Characteristics<br>4.4 Validity Checks and Test Case Generation<br>4.5 Testing Strategies: Unit Testing, Integration Testing, System Testing, Validation Testing  | <b>[15%]</b>  |
| <b>5 Testing Strategies</b><br>5.1 Unit Testing versus Integration testing  | <b>[20%]</b>  |

- 5.2 Unit Testing : Structural testing versus Functional Testing, Types
- 5.3 Structural Test Approach: Basis Path, Data flow and Loop Test
- 5.4 Functional Test Approaches: Equivalence class and Boundary value Analysis, Decision table
- 5.5 Integration Testing: Types and Usage, Designing Stubs and Drivers
- 5.6 System Testing : Types
- 6 **Test Automation and Tools** [15%]
- 6.1 Manual Testing versus Test Automation
- 6.2 Automated Test Data Generation
- 6.3 Test Report Generation and Analysis
- 6.4 Test Result Validation and Verification with Software Requirement

**Course Units and Course Outcomes Mapping:**

Unit No.	Unit	Course Outcomes					
		C01	C02	C03	C04	C05	C06
1	Software Engineering Fundamentals	✓					
2	Process Modeling		✓				
3	Software Design			✓			
4	System Testing				✓		
5	Testing Strategies					✓	
6	Test Automation and Tools						✓

**Programme Outcomes:**

- P01: Ability to understand the concepts of key areas in computer science.  
P02: Ability to design and develop system, component or process as well as test and maintain it so as to provide promising solutions to industry and society.  
P03: Effective communication and presentation skill.  
P04: Ability to understand professional and ethical responsibility.  
P05: Recognition of the need for life-long learning.

**Programme Outcomes and Course Outcomes mapping:**

Programme Outcomes	Course Outcomes					
	C01	C02	C03	C04	C05	C06
P01	✓	✓			✓	✓
P02			✓			
P03		✓	✓			
P04	✓	✓				✓
P05					✓	
P06				✓		
P07				✓		✓

**Text Books/Material References :**

1. Pressman, R. S. -Software Engineering: A Practitioner's Approach -McGraw Hills. (PR)
2. Yogesh Singh- Object Oriented Software Engineering –PHI (YS)
3. Kendall K, Kendall J- System Analysis and Design- PHI (KK)

## Course Curriculum Execution Guidelines

**Semester Objectives:** Enhance reading skill, technical writing skill, communication skill Promote class participation.

**Content Delivery:** The course content shall be delivered by following pattern, wherein teacher must give approximately 75% hours exclusively for imparting conceptual knowledge. Rest 25% hours for demonstration/hands-on regarding supported tool and technology.

Activity Type	Activity	Objective	During
Bridge	Introduction session on a project development life cycle. Discussion and Recollect concept of system which they were done in previous semester.	To fulfil pre-requisites of the course prior to beginning of the semester, at the beginning of the semester	1 <sup>st</sup> two days of semester start.

**Activities/Practicum By Teacher:** *The following activities shall be carried out by the teacher:*

Programme Outcome	Course Outcome	Mode of Transaction	Activity List and Description	Unit	Week	Semester Objective
P02,P03,P05	C05,C06	Demonstration	Demonstration of testing tools	5,6	12,13	Enhance reading skill, technical writing skill, communication skill

**Activities/Practicum By Student:** *The following activities shall be carried out by the students:*

Programme Outcome	Course Outcome	Activity List and Description	Unit	Week	Semester Objective
P02,P03,P04	C04	Role plays on requirement engineering and its phases.	2,4	3,10	Enhance reading skill, technical writing skill, communication skill

**Note:** In addition to the above list course teacher must include those activities that are designed to enrich learning experience during class room teaching, to fulfil semester objective and programme objective.

**Concept linkage:** A course must establish concept linkage as mentioned in the table through content delivery mode, active learning activities or through assessments.

Unit/Sub-Unit	Prior concept linkage	Post concept linkage
1-1.2,2-2.1,2.2.2.3	060060311: Information System Analysis and Design: Unit 2,3 060060113: Computer Fundamentals and Organization : Unit 2,6	
5-5.1,5.2,5.3,5.4	060060114 : Introduction to Web Design	

- The concepts mentioned in **Prior Concept Linkage** shall be taken as base and revised in context of the respective unit/sub-unit by the course teacher.
- The concepts mentioned in **Contemporary Linkage** shall be correlated by the course teacher during the discussion of the respective unit/sub-units with those in current semesters.
- The concepts mentioned in **Post Concept Linkage** shall be correlated by the course teacher during the discussion of the respective unit/sub-units.

### Assessment Pattern

Bloom's Category	% weightage in CIE		
	Quiz	Unit Test	Internal
Remembering	20	30	20
Understanding	60	40	40
Analysis	20	30	40

Subject to change based on the nature of course and as prescribed by Examination Committee of the institute after due approval from Director.

#### Course Level Assessment Questions

Course Outcomes	Questions satisfying achievement of specific CO	Bloom's Taxonomy
CO1	State two points of difference between system analysis and designing phase giving an example.	Remembering
CO2	<p>Differentiate between release planning and iterative planning.</p> <p>Write the effect of combining two process models together. Mention one benefit and one limitation of doing such integration.</p> <p>List all the principles applied to agile culture of software development. Explain any five with examples.</p>	<p>Understanding</p> <p>Understanding</p> <p>Remembering</p>
CO3, CO4	<p>Which verification technique is informal and applicable for verifying small size documents?</p> <p>Why one need to maintain high cohesion and low coupling between modules? Which type of coupling and cohesion are considered best among all?</p>	<p>Understanding</p> <p>Remembering</p>
CO5, CO6	<p>Write any one importance of system testing to achieve quality software.</p> <p>Consider the following "binary search" algorithm. Inputs: n, the length of the following array. A, an integer array with entries A[1], ..., A[n] such that <math>A[i] &lt; A[i+1]</math> for i between 1 and n-1 key, an integer Outputs: index, an integer between 0 and n such that if index = 0 then key does not equal any entry of the array A if index is between 1 and n then <math>A[index] = key</math>  Local Variables: low, medium, high: integers found: boolean</p> <ol style="list-style-type: none"> <li>1. found := false</li> <li>2. low := 1</li> <li>3. high := n</li> <li>4. while ((low is less than or equal to high) and not found) do</li> <li>5.   medium := floor((low + high)/2)</li> <li>6.   if <math>A[medium] = key</math> then</li> <li>7.     index := medium</li> <li>8.     found := true</li> <li>9.   else</li> <li>10.    if <math>A[medium] &lt; key</math> then</li> <li>11.     low := medium + 1</li> <li>12.    else</li> <li>13.     high := medium - 1</li> <li>14.    end if</li> <li>15.   end if</li> <li>16.   end while</li> <li>17.   if not found then</li> <li>18.     index := 0</li> <li>19.   end if</li> </ol> <p>1. Construct a flow graph for this program.  2. What is the cyclomatic complexity of this program?</p>	<p>Understanding</p> <p>Analysis</p>