



**Course description:**

The Course focuses on an approach to software development based on extensive use of pre-existing standard (or customizable) components. It also illustrates how a repository of reusable candidate components can be integrated into a typical evolutionary process model. The Component-based Software Engineering process involves identifying candidate components; qualify each component interface, and adapting components.

**Aims of the course:**

- Describe the role of Component Based Software Engineering (CBSE) within the software life cycle.
- Apply key elements and common methods for CBSE.
- Describe, Compare, contrast and evaluate structured, Object Oriented, data Oriented and formal approaches to component modeling.
- Do some develop CBSE.
- Conduct a review of CBSE requirements and using best practices to determine the quality of a CBS.
- Demonstrate the capacity to use a range of software tools in support of CBS.

**Intended Learning Outcomes: (ILOs)**

**A. Knowledge and Understanding**

- A1) List the basic CBSE Standards and structures.
- A2) List the concept of CBSE Process, and main models.
- A3) List advanced concepts of CBSE modeling, analysis, , prototyping, etc.
- A4) List the concept of Quality of CBSE.

**B. Subject-specific skills**

- B1) Implement solutions of a range of software tools in support of the CBS
- B2) Implement solutions using web service examples and C#

**C. Critical-Thinking Skills**

- C1) Analyze and compare different CBSE models.

**D. General and Transferable Skills (other skills relevant to employability and personal development)**

- D1) Discuss and work in a group in order to design and implement CBS.
- D2) Discuss and work in a group in order to study and present CBS cases

**Course structures:**

Week	Credit Hours	ILOs	Topics	Teaching Procedure	Assessment methods
1	3	A1	<ul style="list-style-type: none"> <li>• History of Component</li> <li>• Component. Definition and essentials</li> <li>• The CBSE processes</li> <li>• Component Charactarsitics</li> </ul>	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation a) Class Participation b) Ist Exam c) 2nd Exam d) Activity file



2	3	A1	<ul style="list-style-type: none"> <li>• Design Diagrams <ul style="list-style-type: none"> <li>○ Activity Diagram</li> <li>○ Use case</li> <li>○ Class Diagram</li> <li>○ Sequence Diagram</li> <li>○ DFD and Context Diagram</li> </ul> </li> </ul>	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation a) Class Participation b) Ist Exam c) 2nd Exam d) Activity file
3,4	4	A1, A3	<ul style="list-style-type: none"> <li>• Component Diagram</li> <li>• Packaging Diagram</li> <li>• Deployment Diagram</li> </ul>	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation a) Class Participation b) Ist Exam c) 2nd Exam d) Activity file
4,5	4	A1,A2, A3	<ul style="list-style-type: none"> <li>• Use of Object Brokers.</li> <li>• DDE</li> <li>• OLE</li> <li>• ActiveX</li> <li>• JavaBeans</li> <li>• Component Anatomy</li> <li>• CORBA</li> <li>• DCOM</li> </ul>	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation a) Class Participation b) Ist Exam c) 2nd Exam d) Activity file
5,6	3	B1, C1	<ul style="list-style-type: none"> <li>• CBSE with reuse (Java, .Net, CORBA)</li> </ul>	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation a) Class Participation b) Ist Exam c) 2nd Exam d) Activity file
7,8,9	3	B2, B3	<ul style="list-style-type: none"> <li>• The reuse landscape</li> <li>• API</li> <li>• Application frameworks.</li> <li>• Design Patterns</li> <li>• Software product lines</li> <li>• product reuse</li> </ul>	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation a) Class Participation b) Ist Exam c) 2nd Exam d) Activity file
10,11	3	B2, B3	<ul style="list-style-type: none"> <li>• Distributed systems issues</li> <li>• Client–server computing</li> </ul>	Presentation methods and techniques,	Diagnostic tests to identify the students level and

			<ul style="list-style-type: none"> <li>Architectural patterns for distributed systems</li> <li>Software as a service</li> </ul>	Sources of information and Instructional Aids	areas of weakness Formal (stage) evaluation a) Class Participation b) Ist Exam c) 2nd Exam d) Activity file
12,13	3	B2, B3	<ul style="list-style-type: none"> <li>Use of Object Brokers.</li> <li>DDE</li> <li>OLE</li> <li>ActiveX</li> <li>JavaBeans</li> <li>Component Anatomy</li> <li>CORBA</li> <li>DCOM</li> </ul>	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation a) Class Participation b) Ist Exam c) 2nd Exam d) Activity file
14,15	5	C2, C3	<ul style="list-style-type: none"> <li>Concepts: Services, SOA, WebServices</li> <li>Services as reusable components</li> <li>Service engineering</li> <li>Software development with services</li> </ul>	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation a) Class Participation b) Ist Exam c) 2nd Exam d) Activity file
15	1	D1, D2	<ul style="list-style-type: none"> <li>Cost/ Benefit Analysis</li> </ul>	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation a) Class Participation b) Ist Exam c) 2nd Exam d) Activity file

## References:

### A. Main Textbook:

Braude, E. J., & Bernstein, M. E. (2016). *Software engineering: modern approaches*. Waveland Press.

### B. Supplementary Textbook(s):

- Component based software engineering, putting the pieces together, Heineman & Council, 1st Ed., pearson, 2001.
- Object-Oriented Software Engineering: Using UML, Patterns and Java, *Bernd Bruegge and Allen H. Dutoit*, 3<sup>rd</sup> Edition, Prentice Hall, 2009.



- Component Software Beyond Object- Oriented Programming, Clemens, Szyperski, Addison Wesley, 2<sup>nd</sup> edition, 2003
- Software Reuse, Architecture, Process and Organization for Business Success, Jacobson M. & Griss P. Jonson, ACM Press, 1998
- Reuse-Based Software Engineering , Hafedh Mili, John Wiley and Sons 2002
- Developing Software with UML: Object- Oriented Analysis and Design in Practice, Bernd Oestereich, Addison Wesley, 2002

#### Assessment Methods:

Methods	Grade	Date
First Exam	20%	
Second Exam	20%	
Assignments (Reports /Quizzes/ Seminar / Tutorials ....)	10%	
Final Examination	50%	

