CPCS-353 Syllabus

Catalog Description

CPCS-353 Software Engineering Practice
Credit: 3 (Theory: 3, Lab: 0, Practical: 1)
Prerequisite: CPCS-351
Classification: Elective

The objective of this course is to further explore the design and thinking of object-oriented software engineering, from analysis through testing. Topics include practices of software project management, project estimation, distributed system architectures (client/ server), distributed object model, building client/server applications based-on Object-oriented technology, and Object-Oriented software engineering approach. The course will also teach students the principles and practices of software testing, validation, verification, maintenance, writing documentation, and evaluation of systems and tools.

Class Schedule

Lab/Tutorial 90 minutes 1 times/week

Meet 50 minutes 3 times/week or 80 minutes 2 times/week

Textbook

Schach, Stephen R., , "Object-Oriented Software Engineering", McGraw-Hill Science/Engineering/Math; 1 edition (2007-09-05)

ISBN-13 9780073523330 **ISBN-10** 007352333X

Grade Distribution

Week	Assessment	Grade %
4	Graded Lab Work 1	2.5
6	Exam 1	15
6	Graded Lab Work 2	2.5
8	Graded Lab Work 3	2.5
12	Exam 2	15
13	Graded Lab Work 4	2.5
14	Lab Exam	10
15	Group Project	20
16	Comprehensive Final Exam	30

Last Articulated

February 20, 2018

Relationship to Student Outcomes

a	b	c	d	e	f	g	h	i	j	k
		x	x							x

Course Learning Outcomes (CLO)

By completion of the course the students should be able to

1. Describe software project management techniques within a team environment. (d)
2. Apply project management tool to assist in the assignment and tracking of tasks in a software development project. (d)
3. **Apply software cost estimation technique for a software system. (k)**
4. **Identify the key features of distributed software systems with particular attention to technical challenges. (c)**
5. Understand the client–server computing model and the layered architecture of client–server systems. (c)
6. **Differentiate between Monolithic and Microservice architectures and discuss how Microservice application revolution powered by the cloud. (c)**
7. Identify software requirements and conduct object-oriented modeling, analysis and design of distributed software systems. (c)
8. **Describe and distinguish among the different types and levels of testing for example unit testing, integration, system (product) testing and acceptance testing. (c)**
9. Describe the role that tools can play in the validation of software (k)
10. **Identify the principal issues associated with software evolution and explain their impact on the software lifecycle. (k)**
11. Discuss about different types of software maintenance and the factors that affect maintenance costs. (k)
12. Explain the problems that exist in achieving very high levels of reliability. (c)
13. Design and model a real client-server application. (k)

Coordinator(s)

Dr. Asif Irshad Khan, Assistant Professor

CPCS-353 Syllabus

Topics Coverage Durations

Topics	Weeks
Project management	2
Software Cost Estimation	1
Distributed software engineering	3
Requirements engineering of Distributed systems	2
System modeling of Distributed systems	2
Software testing	2
Software Evaluation	2
System Maintenance	1