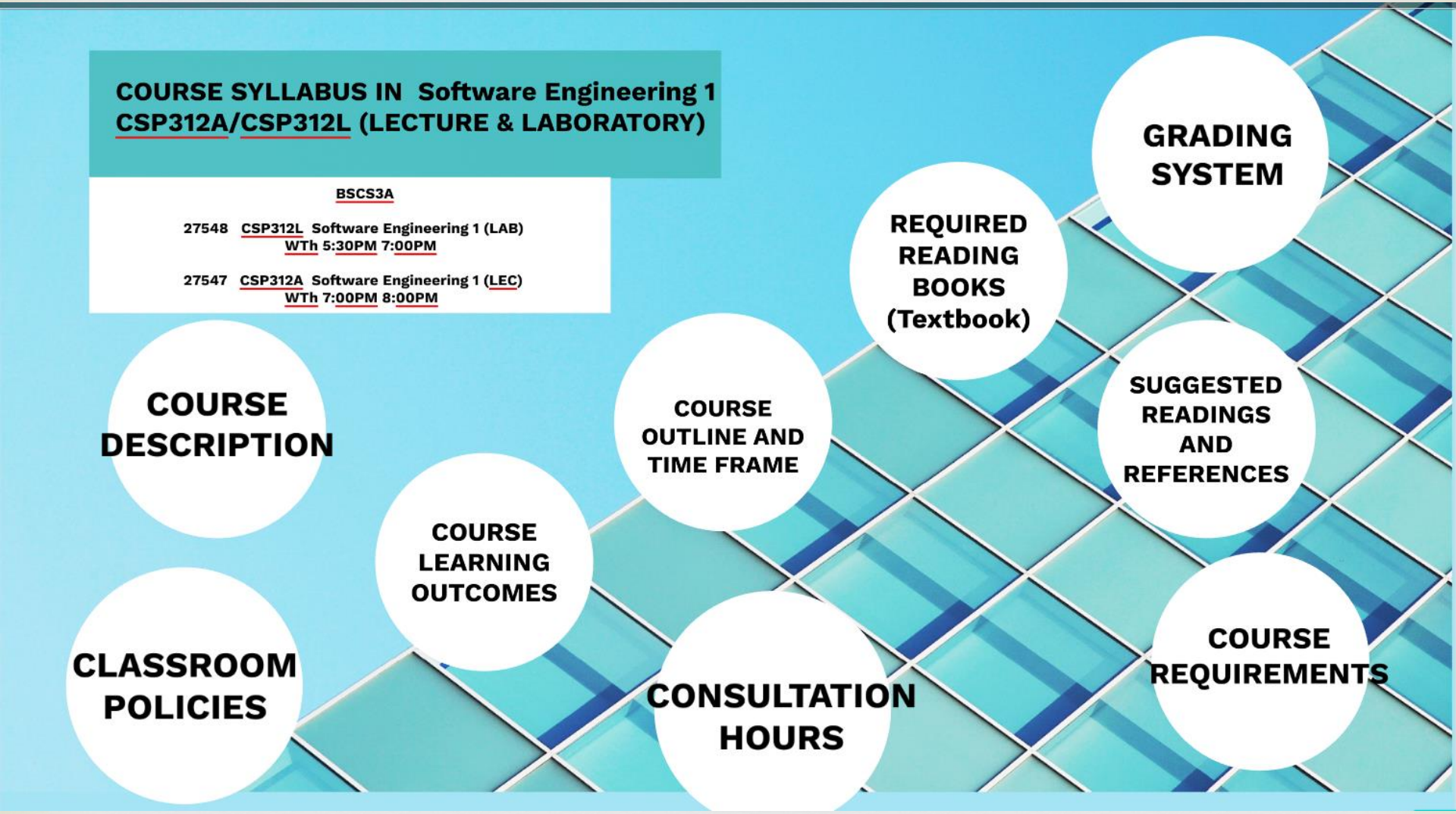
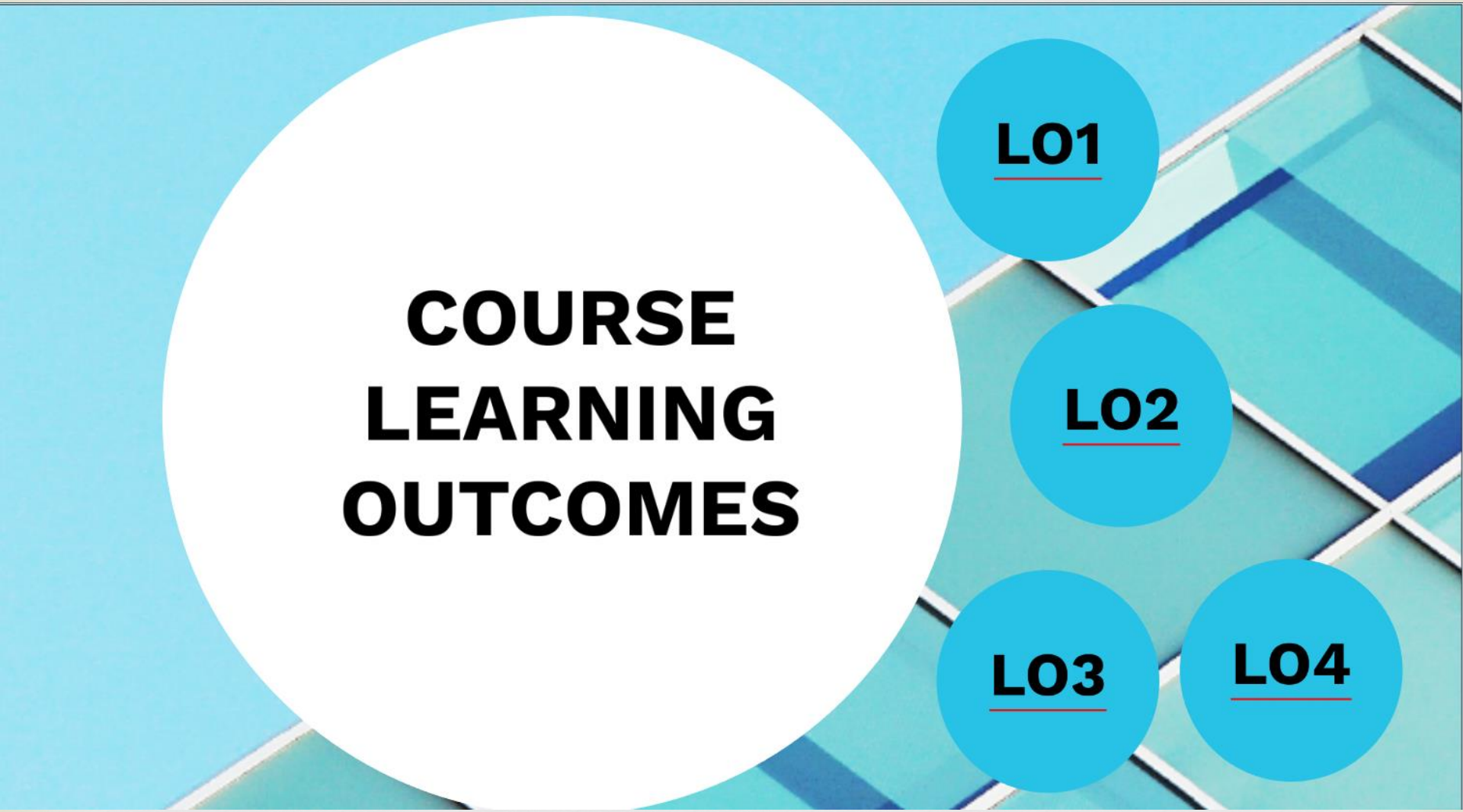

COURSE SYLLABUS IN Software Engineering 1
CSP312A/CSP312L (LECTURE & LABORATORY)
BSCS3A
27548 CSP312L Software Engineering 1 (LAB)
WTh 5:30PM 7:00PM
27547 CSP312A Software Engineering 1 (LEC)
WTh 7:00PM 8:00PM
GRADING SYSTEM
REQUIRED READING BOOKS (Textbook)
COURSE DESCRIPTION
COURSE OUTLINE AND TIME FRAME
SUGGESTED READINGS AND REFERENCES
COURSE LEARNING OUTCOMES
CLASSROOM POLICIES
CONSULTATION HOURS
COURSE REQUIREMENTS

COURSE DESCRIPTION

This course provides an overview of software engineering as a discipline concerned with the application of theory, knowledge, and practice for effectively and efficiently building software systems. It introduces the students to the fundamental principles and methodologies of software engineering. It focuses on the concepts and principles of software requirements engineering, its tools, techniques, and methods for modeling software systems. Various approaches to requirements analysis and review activities are examined.

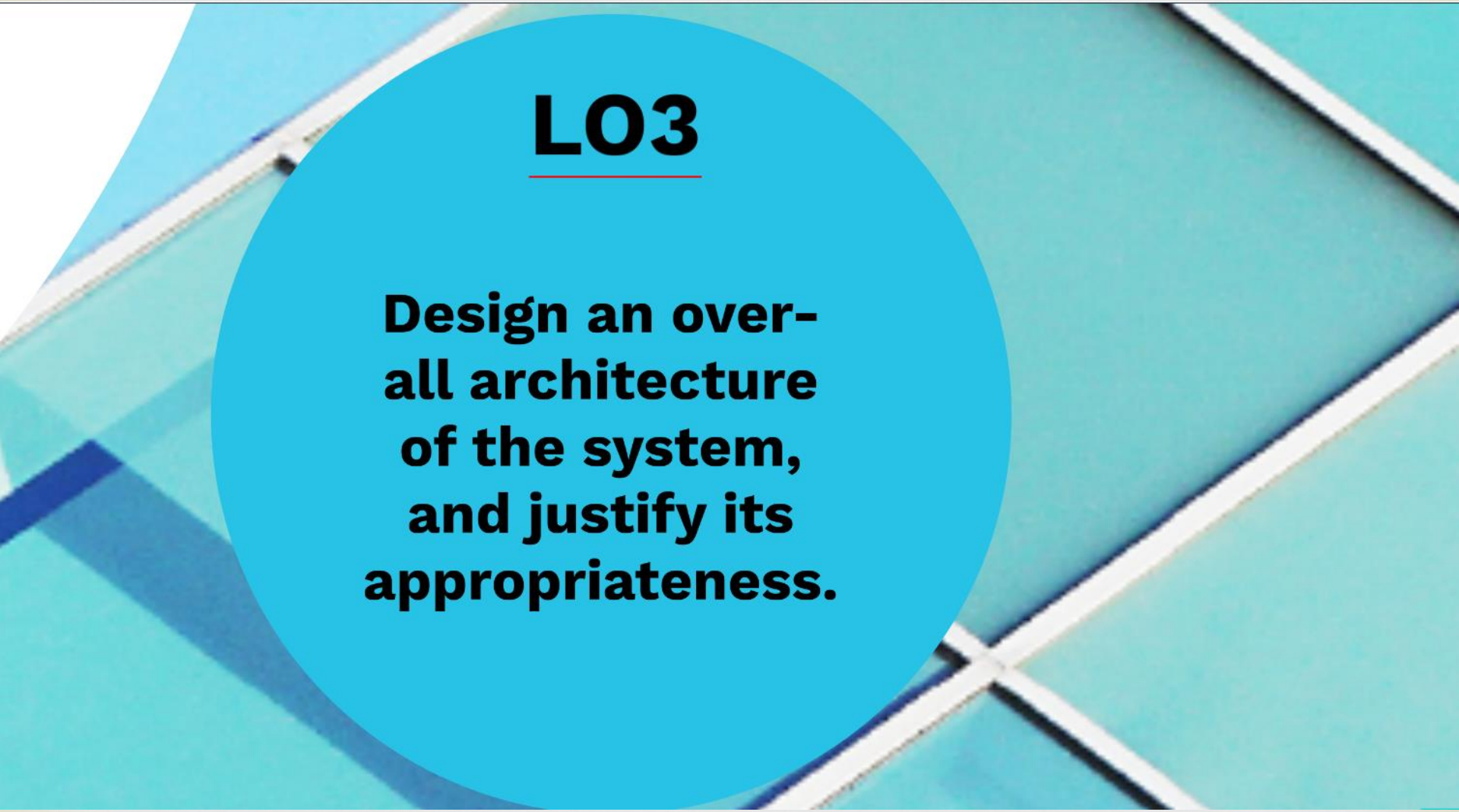

COURSE LEARNING OUTCOMES
LO1
LO2
LO3
LO4

LO1

Discuss the difference between the waterfall-based models and agile-based models, and identify the strengths and weaknesses of these methods.

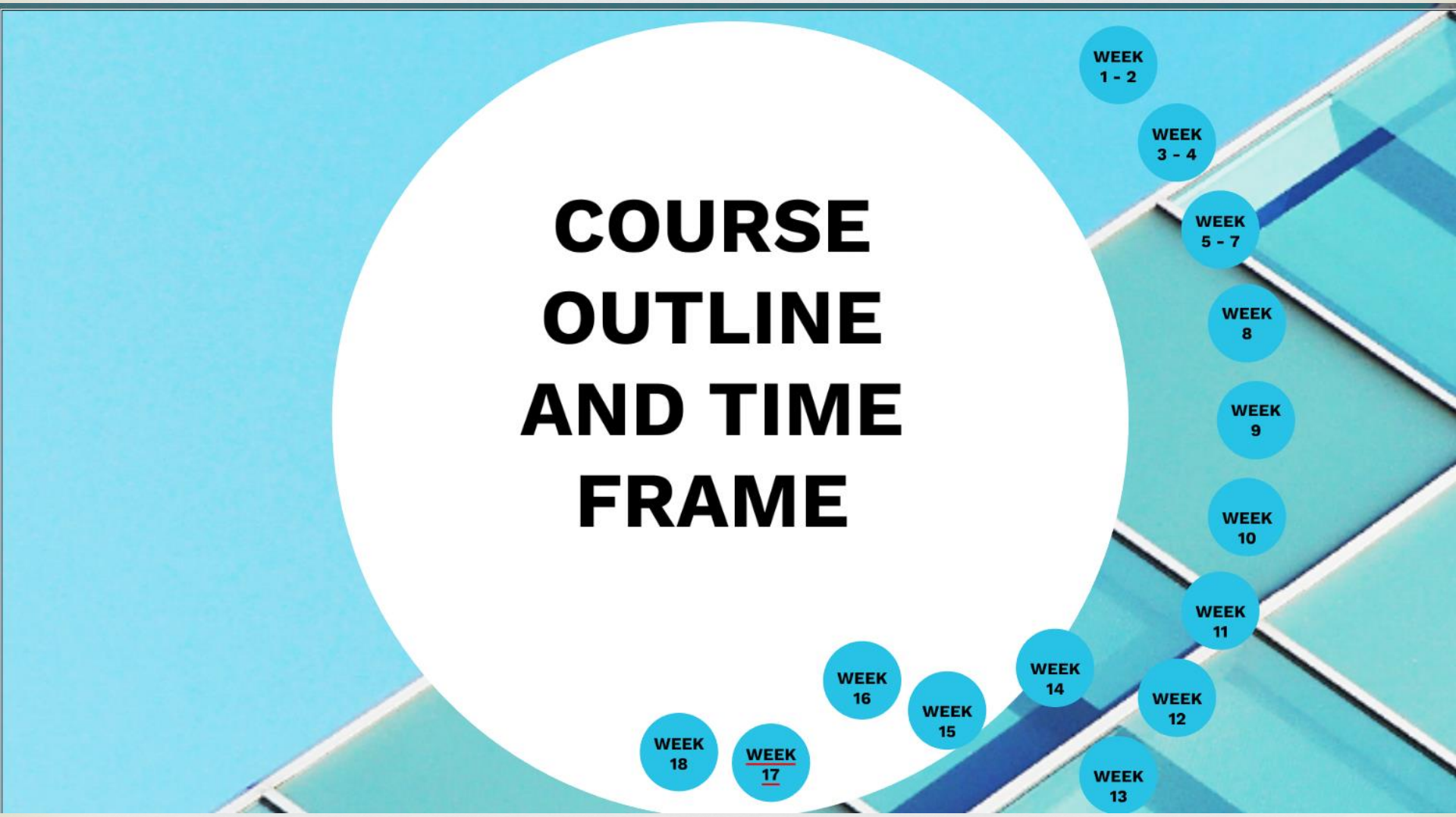
LO2

Extract user-requirements translate these to formal models and present these using UML-based visualizations.


LO3
Design an over-all architecture of the system, and justify its appropriateness.

LO4

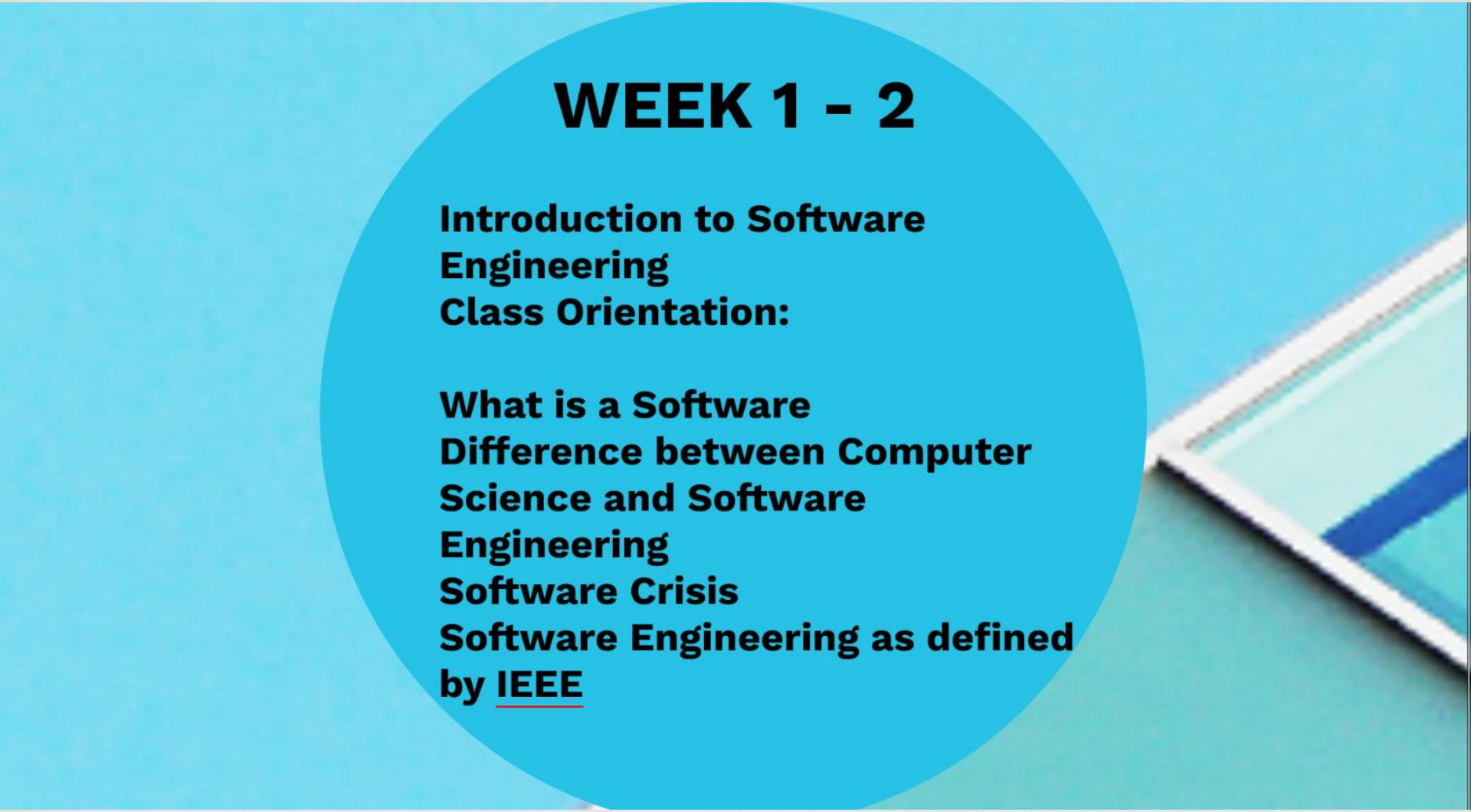
Translate program designs and specifications into actual program codes.


COURSE OUTLINE AND TIME FRAME
WEEK 1 - 2
WEEK 3 - 4
WEEK 5 - 7
WEEK 8
WEEK 9
WEEK 10
WEEK 11
WEEK 12
WEEK 13
WEEK 14
WEEK 15
WEEK 16
WEEK 17
WEEK 18

WEEK 1 - 2

Introduction to Software Engineering
Class Orientation:

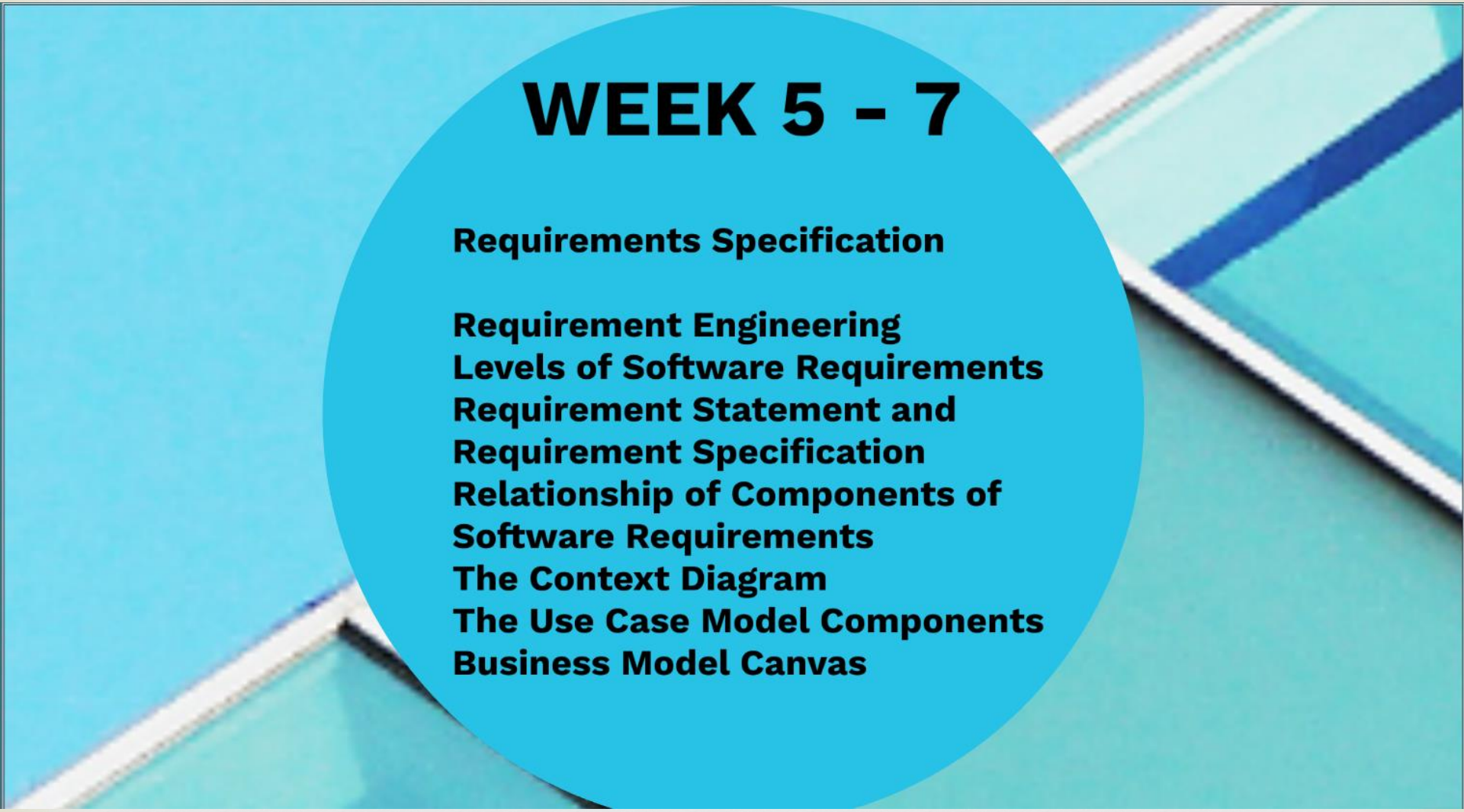
What is a Software
Difference between Computer Science and Software Engineering
Software Crisis
Software Engineering as defined by IEEE



WEEK 3 - 4

Introduction to Software Development

Software Development
Software Engineering Framework
Software Development Loop
Software Engineering Phases
Introduction to Agile Development



WEEK 5 - 7

Requirements Specification

Requirement Engineering
Levels of Software Requirements
Requirement Statement and Requirement Specification
Relationship of Components of Software Requirements
The Context Diagram
The Use Case Model Components
Business Model Canvas

WEEK 8

Project Presentation: Software Requirements Specification

(Learners will be able to present initial overview of the final Project.)

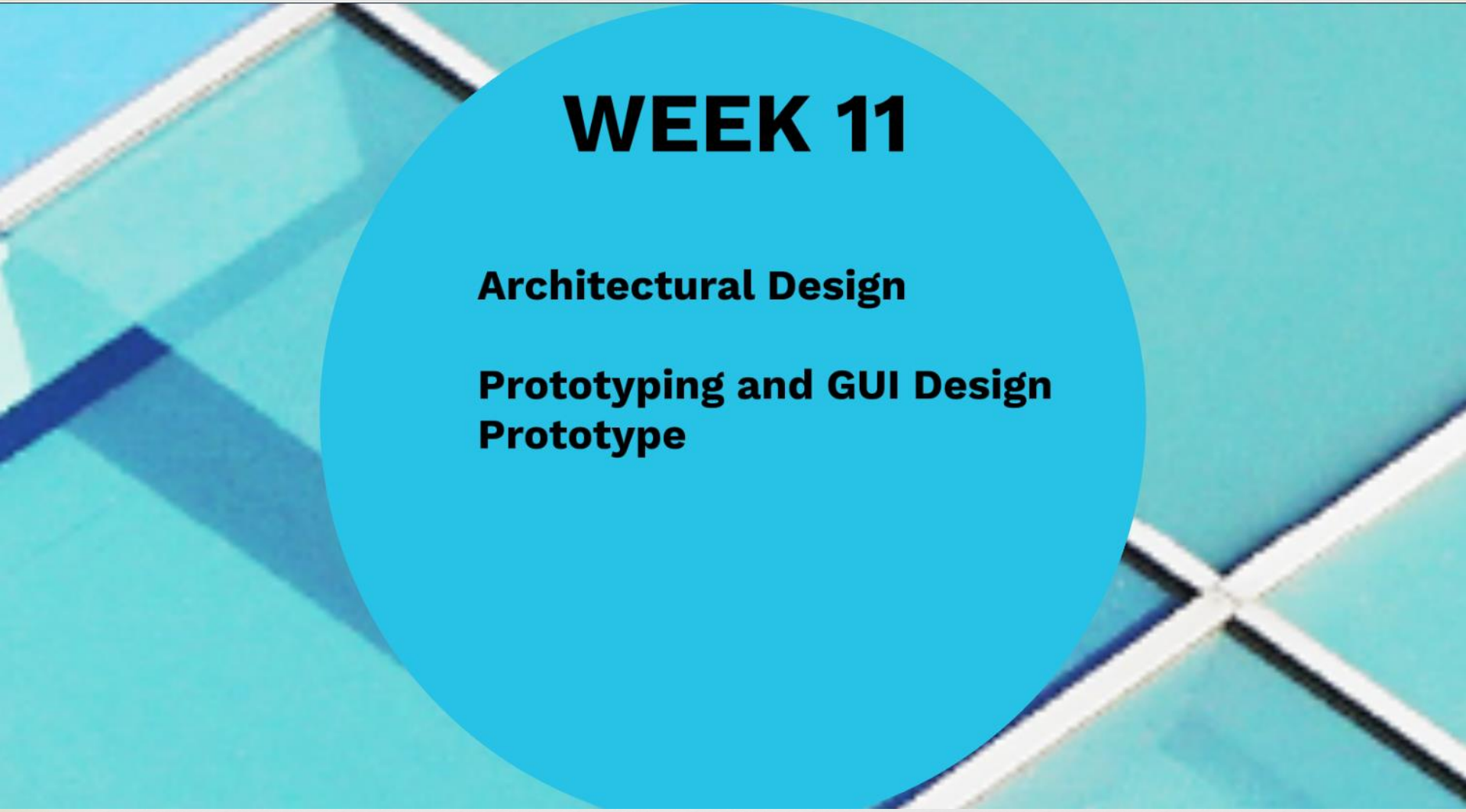
WEEK 9

Midterm Examination

WEEK 10

System Modeling (using UML)

Source and Sink Analysis
Process Models
State Transition Diagram
Data Flow Models
Typical Process
The Notation


WEEK 11
Architectural Design
Prototyping and GUI Design
Prototype

WEEK 12

Introduction to Software Design (HCI)

Software Design
Introduction to Object Oriented Analysis and Design
Derivation of the Object Model – The Coad Methodology

WEEK 13

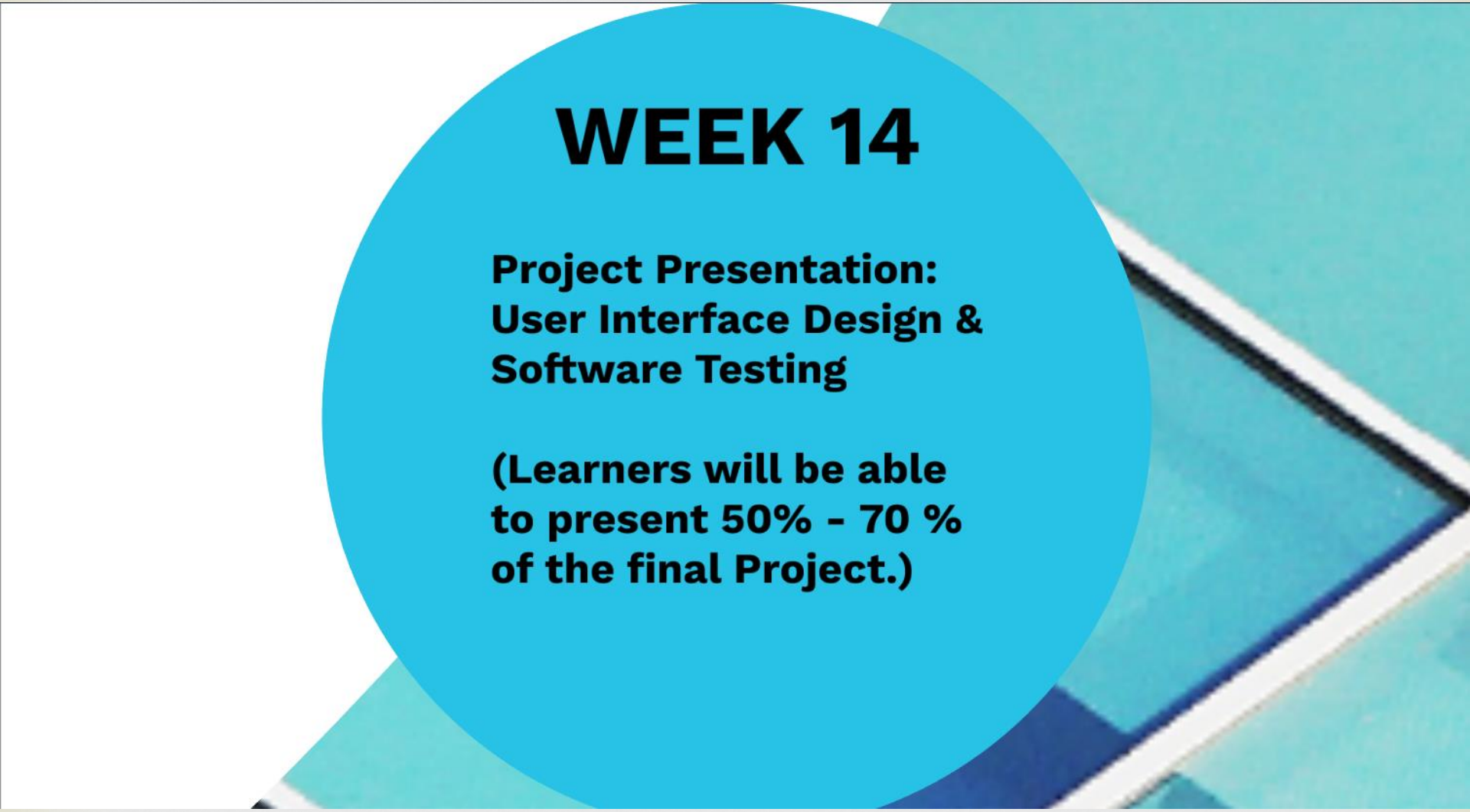
Software Testing

Good Programming Practices and Guidelines
Software Verification and Validation
Testing Versus Development
Unit Testing
Inspection Versus Testing
Debugging

WEEK 14

Project Presentation:
User Interface Design &
Software Testing

(Learners will be able
to present 50% - 70 %
of the final Project.)



WEEK 15

Introduction to Project Management

Identify Structures
Interaction Diagrams – depicting the dynamic behavior of the system
Software and System Architecture
Architectural Views
Architectural Models

WEEK 16

Ethical Issues in Software Development

Why We Need to Address Ethical Issues In Software Engineering

Ethical Challenges Faced by Software Developers

WEEK 17

Final Project Final Presentation, Demonstration and Submission of Deliverables

(Learners will be able to present 100% working final Project and will be able to provide the project documentation.)

WEEK 18

Final Examinations

REQUIRED READING BOOKS (Textbook)

Software Engineering 1 Module

SUGGESTED READINGS AND REFERENCES

- Software Engineering 10th Edition, Ian Somerville, Pearson 2012
- Software Engineering: A Practitioner's Approach, McGraw Hill, 2015
- Software Engineering 3rd Edition, Microsoft Press, 2013
- https://www.genrica.com/vustuff/CS504/CS504_handouts_1_45.pdf
- https://vulms.va.com.pk/categories/cs504-software-engineering-i
- TR-Agile-Development-PSITE
- http://agilemanifesto.org/
- http://agilemanifesto.org/principles.html
- https://simpleprogrammer.com/ethical-issues-software-engineering/

COURSE REQUIREMENTS

Quizzes, Group Work, Case Analysis, Blogging, Project System Output, Examinations

GRADING SYSTEM

Each Midterm and Final Term shall be distributed with a weight of 50.0%. The Final Grade shall be obtained by adding the 50% of the Midterm Grade and 50% of the tentative grade in the final term or computed as:

50% (Midterm Grade) + 50% (Tentative Final Term Grade) = 100% (Final Grade)

Requirement/Assessment Task	Maximum Points
Major Exams	30%
Homework/Activity	20%
Recitation	20%
Quizzes/ Projects	30%
TOTAL	100%

Rubric for any given activity

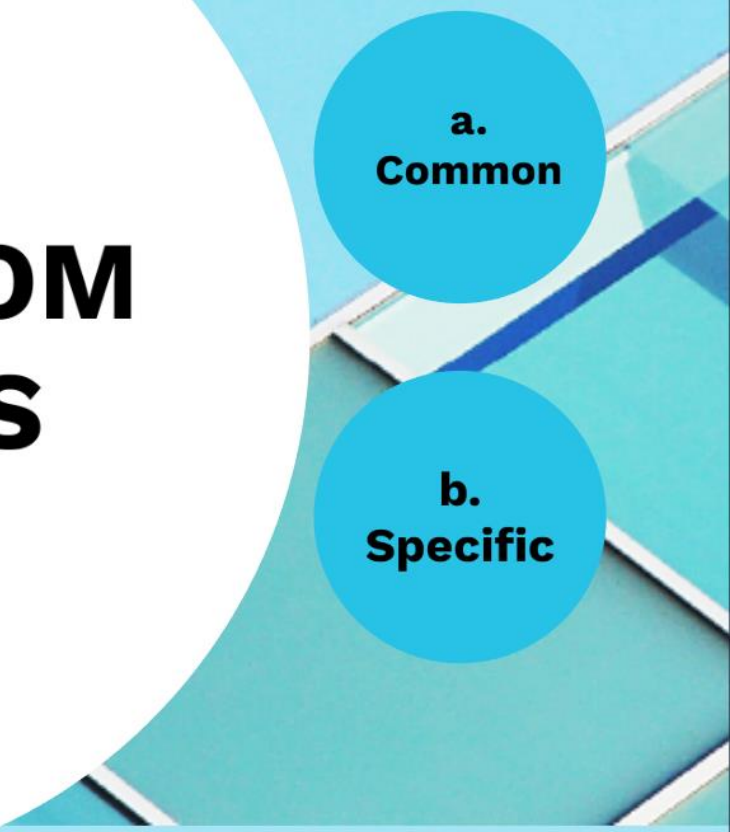
Rubric for pitching, presentation and demonstration activity

Rubric for Final Project

CLASSROOM POLICIES

a. Common

b. Specific



A.
COMMON

1. Attendance checking will be after 15 minutes of the time period, beyond this student will be considered late, 2 accumulated late will be equivalent to 1 absence.

2. Submission of assignments, projects and any take home activities must meet the deadline as specified. Failure to do so will deduct 1 point per day for late submission and will not be accepted after a week.

B.
SPECIFIC

1. The student may copy or download lectures and presentations from the FB group that will be created by the professor.

2. Non-submission of final project and completion of final exam on the specified date will incur an incomplete grade.


CONSULTATION
HOURS
Fri 9:00AM – 6:00PM

Ms. Mayer Z. Sanchez

Maam Yheng

CCS Instructor

ACT Program Coordinator

STAY AT HOME

STAY SAFE

THANK YOU

&

GODBLESS