National University of Sciences & Technology (NUST)
School of Electrical Engineering and Computer Science (SEECS)
Department of Computing

SE-312 Software Construction

Course Code:	SE-312	**Semester:**	6th
Credit Hours:	3+1	**Prerequisite Codes:**	
Instructor:	Fahad Ahmed Satti	**Class:**	BESE-5-AB
Office:	B 207, IAEC	**Telephone:**	
Lecture Days:	To be shared by ACB	**E-mail:**	Fahad.satti@seecs.edu.pk
Class Room:	To be shared by ACB	**Consulting Hours:**	Monday 3-5pm or by appointment
Lab Engineer:	Maryam Sajjad	**Lab Engineer Email:**	Maryam.sajjad@seecs.edu.pk
Knowledge Group:	Software Engineering	**Updates on LMS:**	Weekly

Course Description:

Software construction 1 as a sub-discipline of software engineering (SE) relates to the fundamental principle and activities of SE to architect, implement and validate software-intensive systems. In principle, the theory and practices of software construction aims to apply systematic and engineering-driven steps for the construction of reliable and robust software systems effectively and efficiently. This course aims to equip the students with state-of-the-art knowledge regarding the theory and practices of software construction that helps them become successful software engineers with skills and knowledge for constructing software-intensive systems. The course introduces a multitude of topics in the context of modern day software engineering to disseminate knowledge that includes but not limited to architectural patterns and styles, software factories, software product lines, component and service-oriented engineering and development, quality and reusability in software construction along with model-driven software engineering and advanced topics like DevOps and Crowd-sourced software construction. Both the human-centric and computer aided software construction theory and practices are fundamental part of this course.

Course Objectives:

The objective of this course is to familiarize students with some of the widely utilized advanced concepts and frameworks in software development. The course will also help the attendees develop problem solving skills and will provide them with a chance to not only come up with solutions but also implement them and achieve a stable and working form using advanced concepts and frameworks in C++, Java & Python. At the end of this course, students should be able to design, implement and reason about solutions.

[1] Software Engineering Construction Review Course: http://www.computer.org/web/education/software-engineering-construction-online-course

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**PLO**	**BT Level***
1. Understand the design patterns for software construction.	1	C-2
2. Analyze various techniques to solve algorithmic and real world problems.	2	C-4
3. Develop applications and tools using various frameworks	3	C-6
4. Understand and apply various code management tools and techniques	11	C-3
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain *Remembering (C-1), Understanding (C-2), Applying (C-3), Analyzing (C-4), Evaluating (C-5), Creating (C-6)*		

Resources:

Text Book:

1. P. Bourque and R.E. Fairley, eds., Guide to the Software Engineering Body of Knowledge, Version 3.0, IEEE Computer Society, 2014; www.swebok.org.

Reference Books:

1. A. Fox, D. Petterson. Engineering Software as a Service: An Agile Approach Using Cloud Computing, 1st Edition, Strawberry Canyon LLC, 2014.
2. McConnell, Steven (2004). Code Complete (2nd ed.). Microsoft Press. *ISBN 978-0-7356-1967-8*.
3. D. Sprott and L. Wilkes. Understanding Service-Oriented Architecture, CBDI Forum, 2004. Microsoft Developer Network, [Online:] https://msdn.microsoft.com/en-us/library/aa480021.aspx
4. M. Eriksson. An Introduction to Software Product Line Development, Proceedings of Umeå's Seventh Student Conference in Computing Science, ISSN- 0348-0542, pp. 26-37, 2003.
5. B. Meyer. Object-Oriented Software Construction, Second Edition, ISE Inc., Santa Barbara, Prentice Hall Professional Technical Reference, ISBN 0-13-629155-4, 1997.
6. I. Gorton. Essential Software Architecture, Second Edition, Springer, 2011
7. P. Donohoe. Introduction to Software Product Lines, Software Engineering Institute, 2014. [Online:] http://resources.sei.cmu.edu/asset_files/Presentation/2014_017_101_423722.pdf

Week	Lecture Topic
01	• Code Optimization (Unit Tests, Debugging, Profiling) • Packaging (Intro to Make, Maven & ant) • Version Control(SVN, GIT)
02	• Problem Solving Techniques • Problem Analysis
03	• Dynamic Programming • Software Construction Technologies
04	• Software Construction Technologies
05	• Intro to Java • Database Access with JDBC • Multi-Threaded Programming with Java
06	**OHT-1**
07	• NoSQL • ORM with Hibernate
08	• ORM with Hibernate
09	• Spring Framework
10	• Web Services (XML/REST)
11	• Intro to Mobile Application Development (Android)
12	**OHT-2**
13	• Context Aware/Location Aware Mobile Application Development (Android)
14	• .Net Framework(Invited talks)
15	• Intro to Data Science • AWS Cloud
16	• Intro to IBM Watson API
17	• Intro to Functional Programming
12	**ESE**

Lab Experiments	
01	YouTube Video manager
02	Design, development and analysis of Matrix Multiplication Algorithms
03	Constraint based software development(open ended lab to develop restaurant reservation system)
04	Creating a data driven software application with Java & MySQL
05	Intro to NoSQL
06	ORM with Hibernate
07	Application development with Spring
08	Mobile Application Development
09	Context Aware Mobile Application Development
10	Location Aware Mobile Application Development
11	Intro to Data Science 1
12	Intro to Data Science 2

Grading Criteria (Tentative):	
	Theory (75%) Assignments: 10% Quizzes: 10% OHTs: 30% Final: 50% **Lab (25%)** Lab Tasks: 70% Class Project: 30%

Tools / Software Requirement:	
	Ubuntu Linux, Java, Spring, MySQL, Android SDK, Python, JBoss, Tomcat, IBM Bluemix, Eclipse/Netbeans

Grading Policy:	
Quiz Policy:	During the course two types of unannounced quizzed will be conducted. A paper based quiz, where conceptual learning will be evaluated and practical exams where students will have to solve a problem within 50 minutes and submit on LMS.
Assignment Policy:	• LMS and GitHub will be used to submit assignments and track submission deadlines. • All submissions must have a description document with Introduction, approach, and How to Run Sections clearly defined. • All Software submissions MUST have their own unit tests.
Lab Conduct:	The labs will be conducted for three hours every week. In most cases, a lab handout will be given in advance. The lab handouts will also be placed on LMS. Every student must submit their own solution on LMS. However, students may also be evaluated by oral viva during the lab.
Plagiarism:	Collaboration and group work is encouraged but each student is required to submit his/her own contribution(s). You must cite and acknowledge all sources of information (including copy-pasted code) in your assignments. Cheating and plagiarism will not be tolerated and will lead to strict penalties including negative marks in assignments which will be adjusted in the final score, as well as referral to the SHOD/Dean for appropriate action(s).
General Class Rules:	• No submission via email will ever be acknowledged or accepted. • Each student will have a total of 5 late days which can be utilized for submitting any assignment or project without a penalty, during the course of the semester. • Students will keep track of late days consumed by them. • At the end of the course, for any late day consumed beyond the 5 late days, a penalty of 2% will be applied to the student's assignment.

Program Learning Outcomes (PLOs)

Program outcomes are the narrower statements that describe what students are expected to know and be able to do by the time of graduation. These relate to the knowledge, skills and attitude that the students acquire while progressing through the program. The program must demonstrate that by the time of graduation the students have attained a certain set of knowledge, skills and behavioral traits, at least to some acceptable minimum level. Specifically, it is to be demonstrated that the students have acquired the following graduate attributes:

(i) **Engineering Knowledge:** An ability to apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

(ii) **Problem Analysis:** An ability to identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

(iii) **Design/Development of Solutions:** An ability to design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

(iv) **Investigation:** An ability to investigate complex engineering problems in a methodical way including literature survey, design and conduct of experiments, analysis and interpretation of experimental data, and synthesis of information to derive valid conclusions.

(v) **Modern Tool Usage:** An ability to create, select and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modeling, to complex engineering activities, with an understanding of the limitations.

(vi) **The Engineer and Society:** An ability to apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice and solution to complex engineering problems.

(vii) **Environment and Sustainability:** An ability to understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

(viii) **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

(ix) **Individual and Team Work:** An ability to work effectively, as an individual or in a team, on multifaceted and /or multidisciplinary settings.

(x) **Communication:** An ability to communicate effectively, orally as well as in writing, on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

(xi) **Project Management:** An ability to demonstrate management skills and apply engineering principles to one’s own work, as a member and/or leader in a team, to manage projects in a multidisciplinary environment.

(xii) **Lifelong Learning:** An ability to recognize importance of, and pursue lifelong learning in the broader context of innovation and technological developments.