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

CS-213 Advanced Programming			
Course Code:	CS-213	**Semester:**	Fall 2018
Credit Hours:	3+1	**Prerequisite Codes:**	CS-212 or Equivalent
Instructor:	Abid Rauf	**Class:**	
Office:		**Telephone:**	
Lecture Days:	To be shared by ACB	**E-mail:**	abid.rauf@seecs.edu.pk
Class Room:	To be shared by ACB	**Consulting Hours:**	through email
Lab Engineer:	Ayesha Asif	**Lab Engineer Email:**	ayesha.asif@seecs.edu.pk
Knowledge Group:	Programming Languages	**Updates on LMS:**	Weekly

Course Description:	
	This course introduces advanced concepts in Programming, leading to better production of software applications. It will cover fundamental concepts in Code Management, enhance problem solving skills, introduce functional programming and frameworks in Java, and enable the attendees to employ dynamic programming to solve optimization problems. This Course will be composed of both theory and practical parts.

Course Objectives:	
	The objective of this course is to familiarize students with some of the widely utilized advanced concepts and frameworks in industry. The course will also help the attendees to 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, and Python. At the end of this course, students should be able to design, implement and reason about solutions.

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**PLO**	**BT Level***
1. Differentiate between functional and imperative paradigms.	1	C-5
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)*		

Mapping of CLOs to Program Learning Outcomes

PLOs/CLOs	CLO1	CLO2	CLO3	CLO4
PLO 1 (Engineering Knowledge)	X			
PLO 2 (Problem Analysis)		X		
PLO 3 (Design/Development of Solutions)			X	
PLO 4 (Investigation)				
PLO 5 (Modern tool usage)				
PLO 6 (The Engineer and Society)				
PLO 7 (Environment and Sustainability)				
PLO 8 (Ethics)				
PLO 9 (Individual and Team Work)				
PLO 10 (Communication)				
PLO 11 (Project Management)				X
PLO 12 (Lifelong Learning)				

Resources:

Text Book:	Java: The Complete Reference by Herbert Schildt, 9th Edition, 2014
Reference Books:	1. C++ How to Program by Paul Deitel and Harvey Deitel, 9th Edition, 2014. 2. Thinking in Java by Bruce Eckel, 4th edition, 2006. 3. Java Development with the Spring framework Wrox Publications, by Rod, Hoeller, Arendsen 4. Python, recommended to consult python documentation
Online Resources:	1. http://bigdatauniversity.com 2. http://www.gotw.ca/gotw/043.htm 3. https://kotlinlang.org/docs/kotlin-docs.pdf 4. https://www.r-project.org/

Week	Lecture Topic	Reading List
01	• Basic Concepts (Reusability, Polymorphic Behavior) • Problem Solving Techniques • Problem Analysis	
02	• Overview of Software Patterns and Implementation (Java/C++ or Python)	
03	• C++ Pointers & Structs • C++ Smart Pointers • Memory Management with C++	
04	• Intro to Java	
05	• Multi-Threaded Programing with Java	
06	• Socket Programming	
07	**OHT-1**	
08	• Database Access (JDBC) • ORM with Hibernate	
09	• Code Optimization (Unit Tests, Debugging, Profiling) • Kotlin Language	
10	• Version Control (SVN, GIT) • Packaging and Logging	
11	• Spring Framework	
12	• Web Services (XML/REST) • Micro Services Architecture	
13	**OHT-2**	
14	• Event-driven Programming	
15	• Messaging Services	
16	• Python Basics • R Language	
17	• Crypto Libraries (OpenSSL)	
18	**End Semester Exam**	

Lab Experiments	
01 (Week-1)	Required Software Installation: JDK, Eclipse including GIT and Maven Packages, GIT, Visual Studio, Python
02 (Week-2)	Design and Implementation of a message passing application between two output consumers using Observer Pattern (Preferably in C++)
03 (Week-3)	Creating a data driven software application with C++
04 (Week-4)	Creating a data driven software application with Java + *Semester Project Ideas Discussion*
05 (Week-5)	Create multithreaded data drive application
06 (Week-6)	Create Client-Server file sharing application
07 (Week-8)	Creating two-tier application using Java (Hibernate)
08 (Week-9)	Modify Lab work 07 with logging features and write test cases. Also required to share code with friends using GIT.
09 (Week-10)	*Semester Project Progress Review*
10 (Week-11)	Creating a data driven software application with Spring
11 (Week-12)	Creating micro-services based data driven software application
12 (Week-14,15)	Use messaging services to exchange messages between micro services created in Lab work 11.
13 (Week-16)	Creating a software application with python
14 (Week-17)	*Semester Project Demo*

Grading Criteria (Tentative):		
	Theory (75%)	
	Assignments:	10%
	Quizzes:	10%
	OHT-1:	15%
	OHT-2:	15%
	Final:	50%
	Lab (25%)	
	Lab Tasks:	70%
	Semester Project:	30%

Tools / Software Requirement:	
	Ubuntu Linux, C/C++, Spring, Python, C++(g++),Tomcat Apache Web server, Tomcat, Eclipse

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. • 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.