Curriculum of PhD Computer Science

Department of Computer Science & Information Technology

University of Sargodha

(Applicable from Spring 2015)
# Table of Contents

PhD(CS) Program’s Rationale .................................................................................................................. 3  
PhD Computer Science Electives ........................................................................................................... 6  
Course Title: Advanced Topics in Formal Methods ................................................................................. 6  
Course Title: Special Topics in Cluster & Grid Computing ..................................................................... 7  
Course Title: Advanced Topics in Parallel Processing ........................................................................... 8  
Course Title: Special Topics in Distributed Computing ........................................................................... 9  
Course Title: Special Topics in Open Source Software ........................................................................... 10  
Course Title: Advanced Research Methods ............................................................................................ 10  
Course Title: Evolutionary Computation ................................................................................................. 12  
Course Title: Special Topics in Machine Learning .................................................................................. 13  
Course Title: Special Topics in Artificial Intelligence .............................................................................. 13  
Course Title: Advanced Optimization Methods ....................................................................................... 14  
Course Title: Special Topics in Data Mining ............................................................................................ 15  
Course Title: Peer-To-Peer Systems ........................................................................................................ 16  
Course Title: Social Network Analysis ....................................................................................................... 16  
Course Title: Ubiquitous Information Interaction ..................................................................................... 17  
Course Title: Evolution and Re-Engineering ............................................................................................ 17  
Course Title: Program Comprehension and Reverse Engineering ............................................................ 18  
Course Title: Software Refactoring ......................................................................................................... 19  
Course Title: Advance Software Architecture .......................................................................................... 19  
Course Title: Natural Language Engineering ............................................................................................ 20  
Course Title: Advance Natural Language Engineering ............................................................................ 22  
Course Title: Lexical Functional Grammar Development .......................................................................... 23
Curriculum

for

PhD Computer Science Program

PhD(CS) Program’s Rationale
PhD is highest degree awarded in an academic discipline and is the most satisfying and rewarding educational experience. Students work closely with a faculty member, performing original research, and tackling challenging and unsolved problems. A PhD is a long, in depth research exploration of one topic. By long we’re typically talking about 3-5 years. By in depth we mean that at the end of the PhD you will be the world expert or close to it in your particular area. You will know more than your advisor about your particular research area. You will know about your research than anyone at your department. By one we mean that by the last couple years of your PhD, you will typically be working on only one narrow problem. The PhD is not about breadth, it is about depth the MS and BS degrees are about breadth not depth. The main requirement in the MS and BS degrees is often a large numbers of courses. A PhD program at CSIT, UOS typically requires 18 credit hours of course work.

Program Goals
The goals of the CS PhD Program are to:

• Provide students with a deep understanding of fundamentals as well as important current applications and research, emphasize learning at the research frontier so they can develop theory, methodology and experimental skills to investigate and solve state of the art technological issues, and be successful professionals.
• Provide technical and system support to dedicated and innovative scholars to continue their cutting-edge research, and participate in the advancement of the discipline.
• Provide ties to alumni to enhance their opportunities for continued learning and leadership, establish liaison with Industry to advise and to provide research opportunities to Temple graduates and the CS Department.

Minimum Criteria for Admission in Ph.D Computer Science

Admission requirement:
1. For admission into the PhD program at CS & IT program, the minimum CGPA 3.0 (out of 4.0 in the Semester System) or First Division (in the Annual System) in M.Phil/M.S/Equivalent degree in the following programs/disciplines:

• MS/MPhil. Computer Science
• MS Software Engineering
• MS Computer Engineering
• MS Information Technology
• MS Information Systems
• MS Information Security
• MS Computer Networks

2. The student must have minimum 6 credit hours of research work with dissertation in MS/M.Phil

**Subject Test:**

A subject test conducted by the department or as decided by the university must be cleared prior to admission for the PhD Program. The student must have a score of 70% in subject test for securing admission in PhD or as per university rules and HEC guidelines.

**Course Work:**

Course work of 18 credit hours preferably in the first year is required to be completed and followed by a comprehensive examination for granting candidacy as PhD researcher.

**Comprehensive Examination**

The student becomes eligible to appear in the Comprehensive Examination after passing the PhD coursework with a minimum CGPA of 3.0 or as per university rules.

**Research Work**

The student can start work on PhD thesis only after passing the Comprehensive Exam, **AND** Acceptance/defense of synopsis by the Graduate Studies Committee.

**PhD Dissertation**

The PhD Dissertation must contain original research work.

**Plagiarism Test**

The Plagiarism Test shall be conducted on the Dissertation before its submission to foreign experts.

**Foreign Expert Evaluation:**
The PhD dissertation must be evaluated by at least two PhD experts from technologically/academically advanced foreign countries in addition to local Committee members or as per university rules and HEC guidelines.

**Open Defense:**

An open defense of Dissertation is essential part of PhD Program after positive evaluation.

**PhD Computer Science Electives**

<table>
<thead>
<tr>
<th>S No</th>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CS-7361</td>
<td>Advanced Topics in Formal Methods</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>2</td>
<td>CS-7521</td>
<td>Special Topics in Cluster &amp; Grid Computing</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>3</td>
<td>CS-7522</td>
<td>Advanced topics in Parallel Processing</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>4</td>
<td>CS-7551</td>
<td>Special Topics in Distributed Computing</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>5</td>
<td>CS-7311</td>
<td>Special Topics in Open Source Software</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>6</td>
<td>SS-7001</td>
<td>Advanced Research Methods</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>7</td>
<td>CS-7821</td>
<td>Evolutionary Computation</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>8</td>
<td>CS-8851</td>
<td>Special Topics in Machine Learning</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>9</td>
<td>CS-8821</td>
<td>Special Topics in Artificial Intelligence</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>10</td>
<td>CS-8852</td>
<td>Advanced Optimization Methods</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>11</td>
<td>CS-8822</td>
<td>Special Topics in Data Mining</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>12</td>
<td>CS-8551</td>
<td>Peer-To-Peer Systems</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>13</td>
<td>CS-7851</td>
<td>Social Network Analysis</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>14</td>
<td>CS-7511</td>
<td>Ubiquitous Information Interaction</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>15</td>
<td>CS-8311</td>
<td>Evolution and Re-Engineering</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>16</td>
<td>CS-8312</td>
<td>Program Comprehension and Reverse Engineering</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>17</td>
<td>CS-8313</td>
<td>Software Refactoring</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>18</td>
<td>CS-8314</td>
<td>Advance Software Architecture</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>19</td>
<td>CS-7840</td>
<td>Natural Language Engineering</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>20</td>
<td>CS-8840</td>
<td>Advance Natural Language Engineering</td>
<td>3(3, 0)</td>
</tr>
<tr>
<td>21</td>
<td>CS-7841</td>
<td>Lexical Functional Grammar Development</td>
<td>3(3, 0)</td>
</tr>
</tbody>
</table>
Course Title: Advanced Topics in Formal Methods

Course Code: CS-7361

Course Structure:
Credit Hours: 3
Prerequisites:

Course Objectives:
The objective of the course is the study of the most important among the qualities of software: correctness. Correctness will be studies looking at the conceptual perspective as well as the realization perspective. Modeling and verification of both static (data) and dynamic (processes) aspects will be considered. The various topics will be treated emphasizing methodological, theoretical and practical facets. The course will introduce various forms of logic (first-order logic, temporal logics, fixpoint logics), techniques and tools for automated verification. After the successful completion of the course the student will have acquired techniques and methods for proving correctness of programs and conceptual models.

Course Syllabus:

First Order Logic and Set theory. Set theory and first order logic form the foundations of mathematics and formal reasoning. We will cover the syntax and semantics of first-order logic, basic model theory, basic proof theory, Godel's completeness theorem, the compactness theorem, the Lowenheim-Skolem theorems, Godel's incompleteness theorem, and non-standard models. We will cover aspects of axiomatic set theory. We will discuss how logic is used to model and reason about computation.

1. Undecidability and Decision Procedures. A long standing goal in mathematics and computer science is to automate proof discovery and mathematical reasoning. We will cover basic undecidability results. We will also discuss recent developments in decision procedures, including the DPLL algorithm for deciding SAT (Boolean Satisfiability) and the SMT (Satisfiability modulo theories) framework.

2. Reactive Systems. Reactive systems are non-terminating systems that interact with an environment. To reason about such systems, we will study various temporal logics and calculi, automata on infinite objects, the notions of safety and liveness, and the theory of refinement.

3. Tools and Applications. We will discuss current verification tools, including interactive theorem provers, SMT solvers, and model checkers. We will discuss applications to hardware verification, software verification, databases, and security.

Reference Material:


• Verification, Model Checking, and Abstract Interpretation By Ranjit Jhala and David Schmidt, 2011.


• Decision Procedures: An Algorithmic Point of View by Daniel Kroening, Ofer Strichman and R.E. Bryant, 2008.

Course Title: Special Topics in Cluster & Grid Computing

Course Code: CS-7521

Course Structure:
Credit Hours: 3

Prerequisites:

Course Objectives:
The purpose of the course is to provide basic knowledge on the most important principles, methods, tools, systems, standards, etc. behind these two evolving basic technologies. Detailed description of the topic of the course is beyond the scope of this article.

Course Syllabus:

Reference Material:
• Distributed and Parallel Systems: In Focus: Desktop Grid Computing by Peter Kacsuk, Robert
Lovas, Zsolt Nemeth.

- Grid Computing Models By D. Janakiram.

Course Title: Advanced Topics in Parallel Processing

Course Code: CS-7522

Course Structure:
Credit Hours: 3

Prerequisites:

Course Objectives:
The major objective of this course is to develop structural intuition of how the hardware and the software work, starting from simple systems to complex shared resource architectures; to provide guidelines about how to write and document a software package; to familiarize the audience with the main parallel programming techniques and the common software packages/libraries.

Course Syllabus:
Different topics related to parallel and systolic computations will be covered at various levels of details. Example of such topics are: architectures of parallel processors and vlsi computational networks, languages and programming environments for parallel systems, the design and analysis of parallel and systolic algorithms, reconfigurable and data driven processor arrays, complexity measures of vlsi computations, and the application of parallel processors to supercomputing.

Reference Material:
- Encyclopedia of Parallel Computing (Springer Reference) by David Padua. 2011

Course Title: Special Topics in Distributed Computing

Course Code: CS-7551

Course Structure:
Credit Hours: 3

Prerequisites:
Course Objectives:
Be able to understand and develop distributed computing systems. Topics include principles of naming and location, atomicity, resource sharing, concurrency control and other synchronization, deadlock detection and avoidance, security, distributed data access and control, integration of operating systems and computer networks, distributed systems design, consistency control, and fault tolerance.

Course Syllabus:
Building correctly functioning, performance-oriented, reliable and secure distributed systems, especially in web-based environments using Java and XML; study the design principles of distributed systems and their application to the modern networked environment; fundamental distributed systems theory including group communication, synchronization, concurrency control, load balancing and scheduling, replication, fault-tolerance, and network security; client/server systems, middleware and middleware-based systems, network computing, and networked file systems.

Topical Areas: The course will examine issues related to replication, fault tolerance, consistency, scalability, isolation and privacy in large-scale distributed systems. The course will include an in-depth coverage of cloud computing strategies to process voluminous data. Other topics include Concurrency and Synchronization, concurrency control, deadlock detection and avoidance, security, distributed data access and control, integration of operating systems and computer networks, distributed systems design, consistency control, and fault tolerance.

Reference Material:
- Distributed Systems: Principles and Paradigms by Andrew S. Tanenbaum and Maarten van Steen

Course Title: Special Topics in Open Source Software
Course Code: CS-7311
Course Structure:
Credit Hours: 3
Prerequisites:
Course Objectives:
Throughout the last decade, the phenomenon of open source software has evolved from what was once a technical curiosity into a mainstay of routine business practice. The emergence of production quality systems such as Linux, MySQL, and Apache have allowed for open source technologies to be viable options for firms choosing to develop their own information technology infrastructure. The distinctive nature of open source communities, firms, and technologies, however, gives rise to issues that are profoundly different from those encountered when a system is acquired from a traditional vendor.
Course Syllabus:
In this course, we will examine the issues associated with open source technologies, with a focus on understanding the implications for businesses that are interested in using them. Through a combination of readings, presentations, discussion and hands-on projects, we will examine: the characteristics of key open source technologies (Linux, MySQL, Apache, et al), the nature of open source communities, their development processes, the evolving structure of the open source industry.

Reference Material:

Course Title: Advanced Research Methods
Course Code: SS-7001
Course Structure:
Credit Hours:
Prerequisites:
Course Objectives:
The primary objective of this course is to develop a research orientation among the scholars and to acquaint them with fundamentals of research methods. Specifically, the course aims at introducing them to the basic concepts used in research and to scientific social research methods and their approach. It includes discussions on sampling techniques, research designs and techniques of analysis.

Course Syllabus:

Reference Material:

- Research Methods by Francis C. Dane, Brooks/ Cole Publishing Company, California.
- Research Methods by Ram Ahuja, Rawat Publications (2001)
- The Computer Science and Engineering Handbook by (Editor-in-Chief) By Allen B. Tucker, jr. CRC Press, A CRC Handbook Published in co-operation with A (only relevant parts of chapters of Chapter-2, Chapter-3, Chapter-4 Chapter-9, Chapter-10 & Chapter-32).

Course Title: Evolutionary Computation
Course Code: CS-7821
Course Structure:
Credit Hours: 3
Prerequisites:
Course Objectives:
Evolutionary Computation can be considered as a sub-field of Artificial Intelligence. Evolutionary algorithms use Nature as a metaphor and are inspired in the principles of natural selection and genetics. These algorithms have been applied successfully for solving difficult problems across a broad spectrum of fields, including engineering, economics and finance, architecture, design, automatic programming, art generation, and many others. In this course, you will learn the basic working principles of these algorithms.

Course Syllabus

Reference Material:


Course Title: Special Topics in Machine Learning

Course Code: 8851

Course Structure: Lecture: 3/ Lab: 0

Credit Hour: 3

Prerequisite: Linear Algebra/ Probability Theory

Course Objectives:
This course covers several modern machine-learning topics at an advanced level. The current list of possible topics includes deep learning, maximum-margin methods, kernel methods, graphical models and Monte Carlo methods. The theoretical and mathematical underpinnings of these topics will be discussed along with their role in modern machine learning by reading and discussing recent publications.

Course Syllabus:

Reference Material:

- Pattern Recognition and Machine Learning by Chris Bishop [Text Book]
- Kernel Methods for Pattern Analysis by John Shawe-Taylor & Nello Cristianini [Text Book]
Course Title: Special Topics in Artificial Intelligence

Course Code: CS-8821

Course Structure: Lecture: 3/ Lab: 0

Credit Hour: 3

Prerequisite: Introduction of Artificial Intelligence

Course Objectives:
The students should obtain advanced theoretical knowledge and technical skills on the topics covered in the course. The course will enable students to carry out advanced research projects on topics within artificial intelligence and apply the knowledge to complex intelligent system design and development. Upon completion of the course the candidate should be able to:

- Discuss advanced theoretical and technical issues in artificial intelligence.
- Apply knowledge for complex intelligent system design and development.
- Carry out advanced research on artificial intelligence.

Course Syllabus:
The course covers advanced theoretical and technical issues in artificial intelligence. It will focus on some selected topics and applications, such as artificial life, agent and multi-agent systems, graphical models, advanced neural networks, genetic algorithms and programming, natural language processing, case-based reasoning, cognitive science, and neuro-computing.

Reference Material:
- Most of the course will be taught by using research paper on topics
Course Title: Advanced Optimization Methods

Course Code: CS-8852

Course Structure: Lecture: 3/ Lab: 0

Credit Hour: 3

Prerequisite: Linear Algebra/Multivariate Calculus

Course Objectives:
This course introduces the principal algorithms for linear, network, discrete, nonlinear, dynamic optimization and optimal control. Emphasis is on methodology and the underlying mathematical structures. Topics include the simplex method, network flow methods, branch and bound and cutting plane methods for discrete optimization, optimality conditions for nonlinear optimization, interior point methods for convex optimization, Newton's method, heuristic methods, and dynamic programming and optimal control methods.

Course Syllabus:
The course takes a unified view of optimization and covers the main areas of application and the main optimization algorithms. It covers the following topics: Linear optimization, Robust optimization, Network flows, Discrete optimization, Dynamic optimization, Nonlinear optimization.

Reference Material:

Course Title: Special Topics in Data Mining

Course Code: CS-8822

Course Structure: Lecture: 3/ Lab: 0

Credit Hour: 3

Prerequisite: Introductory Data Mining Course

Course Objectives:
Data Mining is one of the hottest fields in Computer Science. Data has been accumulating throughout the computer age in many forms, including database systems, spreadsheets, text files, and recently web pages. Data mining aims to search through data for hidden relationships and patterns in your data. This is a special topic course on data mining. We will cover advanced topics such as large-scale data mining using Map-reduce, similarity search (including minwise hashing and locality sensitive hashing), mining data streams, mining social networks, relational data mining, and matrix factorization methods for data
mining. This course will be highly beneficial to students whose research interests are in database, data mining, bio-informatics, information retrieval, decision science and artificial intelligence, and also to those who may need to apply data mining to any application.

Course Syllabus
This is a seminar course that will focus on recent developments of advanced data mining techniques and their applications to various problems. After the introductory lectures, subsequent classes will mainly based on research papers. Topics will include: Large-scale Data Mining, Mining Data Streams, Mining Social Networks, Relational Data Mining, Tree/Graph Mining, Privacy-preserving Data Mining, High-Dimensional Data Clustering, Web Applications (including advertising, recommendation, and summarization).

Reference Material:
- Ethical Data Mining Applications for Socio-Economic Development by Hakikur Rahman, Isabel Ramos, 2013.
- The latest research articles on the topics will be presented.

Course Title: Peer-To-Peer Systems

Course Code: CS-8551

Course Structure: Lecture: 3/ Lab: 0

Credit Hour: 3

Prerequisite:

Course Objectives:
Peer-to-peer systems have recently gained a lot of attention in the social, academic, and commercial communities. One of the early driving forces behind the peer-to-peer concept is that there are many PCs in homes and offices that lie idle for large chunks of time. This course will introduce the state of the art knowledge about peer to peer systems.

Course Syllabus:
Overview of P2P Systems and brief history; Taxonomy of P2P Networks/Systems and Analysis of popular P2P Systems; Analysis of unstructured P2P Systems; Analysis of structured P2P Systems; Search Efficiency; P2P-based content delivery; Security and Reliability; Replication in peer-to-peer systems; Anonymity in peer-to-peer systems; Social, Legal and Privacy aspects of P2P Systems.

Reference Material:
• The latest research articles on the topics will be presented.

Course Title: Social Network Analysis

Course Code: CS-7851

Course Structure: Lecture: 3/ Lab: 0

Credit Hour: 3

Prerequisite: Introductory Data Mining Course

Course Objectives:
The main learning objective with this course is to enable students to put Social Network Analysis projects into action in a planned, informed and efficient manner. This overarching goal involves the following subtasks:

Formalize different types of entities and relationships as nodes and edges and represent this information as relational data; Plan and execute network analytical computations; Use advanced network analysis software to generate visualizations and perform empirical investigations of network data; Interpret and synthesize the meaning of the results with respect to a question, goal, or task; Collect network data in different ways and from different sources while adhering to legal standards and ethics standards.

Course Syllabus:
Introduction to social networks; random network models; identifying connected components; giant component; average shortest path; diameter; preferential attachment; network centrality; betweenness; closeness; clustering; community structure; modularity; overlapping communities; small world network models; contagion; opinion formation; applications of social network analysis; social media networks.

Reference Material
• Social Network Analysis: A Handbook by John Scott, 2000
• Models and Methods in Social Network Analysis by Peter J. Carrington, John Scott, Stanley Wasserman, 2005.
• Social Network Analysis: History, Theory and Methodology by Christina Prell, 2011.
• The latest research articles on the topics will be presented.
Course Title: Ubiquitous Information Interaction

Course Code: CS-7511

Course Structure: Lecture: 3/ Lab: 0

Credit Hour: 3

Prerequisite: Introductory Data Mining Course

Course Objective:
The purpose of this course is to introduce students with: the objectives and the historical development of the field of ubiquitous computing; development in new materials; fundamentals of sensor technology and sensor networks; design of new (often embedded) interactive artefacts; context aware and adaptive systems; middleware for fine-grained distributed systems; analysis and coordination of complex systems; new styles of interaction, e.g. tangible interfaces; most important applications in the field; general implications of the field.

Course Syllabus:
Information Interaction; Seminal ideas of ubiquitous computing; Tangibility and Embodiment; Social computing; Privacy; Critical and cultural perspectives; Mobility and Spatiality; Mobile Technology in the Messy Now; Infrastructure; Seams, seamlessness, seamfulness; Evaluating Interaction of Ubicomp systems.

Reference Material:
• Ubiquitous Computing: Smart Devices, Environments and Interactions, By Stefan Poslad. 2011.
• The latest research articles on the topics will be presented.

Course Title: Evolution and Re-Engineering

Course Code: CS-8311

Course Structure: Lecture: 3/ Lab: 0

Credit Hour: 3

Prerequisite:

Course Objectives:
The objective of the course is to examine the set of tools and techniques that allow a smooth re-engineering of old legacy systems to new more maintainable systems that use more appropriate and robust advanced technologies (i.e. Object-Oriented, Network-Centric).

**Course Syllabus:**

Introduction: Challenges of evolution, Legacy systems; Evolution Process: Laws of software evolution, Evolution models, Testing in the context of evolution, Metrics for evolution; Evolution Activities: Concepts of, and techniques for activities e.g. Reverse engineering, Re-factoring, Program Transformation, Visualization; Re-engineering techniques: Code restructuring, Source code analysis, Architecture Recovery; Topics in re-engineering research.

**Reference Material:**

- Aspect-Oriented Software Development by Filman, Elrad, Clarke, Aksit, Addison-Wesley ISBN 0321219767, 2004
- The latest research articles on the topics will be presented.

**Course Title: Program Comprehension and Reverse Engineering**

**Course Code:** CS-8312

**Course Structure:** Lecture: 3/ Lab: 0

**Credit Hour:** 3

**Prerequisite:**

**Course Objectives:**

The objective of this course is to introduce students with reverse engineering techniques and algorithms.

**Course Syllabus:**

Static Analysis: Parsing, lexical analysis, issues in parsing languages; Program analysis: Control flow analysis, Data flow analysis, flow graphs, program dependence graphs, call graphs; Dynamic analysis: Profiling, dynamic testing; Reverse engineering: Design recovery and re-documentation, challenges in reverse engineering, reverse engineering approaches; Reverse engineering techniques: Techniques for reverse engineering at the program level, Techniques for reverse engineering at the architectural level
Reverse Engineering tools.

**Reference Material:**

- Reverse Engineering, by Linda Wills, Philip Newcomb. 1996.
- Reversing: Secrets of Reverse Engineering by Eldad Eilam 2011.

**Course Title: Software Refactoring**

**Course Code:** CS-8313

**Course Structure:** Lecture: 3/ Lab: 0

**Credit Hour:** 3

**Prerequisite:**

**Course Objectives:**

The objective of this course is to learn the techniques for: Use refactoring to facilitate adding new functionality to system; Use refactoring to improve design; Refactor existing applications to make them more maintainable; Use tests during refactoring; Recognize when and when not to refactor; Identify and choose the appropriate type of refactoring technique to solve specific problems.

**Course Syllabus:**

Refactoring principles: Reasons for refactoring, what to refactor, Challenges in refactoring ; Refactoring categories: Refactoring in the small and large; Refactoring techniques: Recognizing bad smells in code, refactoring for organizing code, higher abstraction, improvement and others; Refactoring of UML models, Tool support for refactoring: Strengths and limitations

**Reference Material:**

- Refactoring: Improving the Design of Existing Code By Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts

**Course Title: Advance Software Architecture**

**Course Code:** CS-8314

**Course Structure:** Lecture: 3/ Lab: 0
Credit Hour: 3

Prerequisite:

Course Objectives:
This course enables the students to understand the challenges of advanced software design and the issues associated with large-scale software architectures, frameworks, patterns and components. To develop the students' understanding of the tools and techniques that may be used for the automatic analysis and evaluation of software.

Course Syllabus:
Re-use in architectures: Software product lines, evaluation and validation of product lines, product line testing, re-use in product lines; Service oriented architectures (SOAs): SOA concepts, risks and challenges, quality attributes and SOAs, evaluating and testing SOAs; Architectural evaluation: Methods for architectural analysis, Comparison of methods; Architectural evolution and reconstruction: Models of software evolution, analysis and metrics for evolution, Techniques and tools for architecture reconstruction; Architectures in dynamic environments: Modeling and analyzing dynamic software architectures; Self healing architectures: The need for self-healing, approaches for self healing and state of art topics in software architecture.

Reference Material:

Course Title: Natural Language Engineering
Course Code: CS-7840
Course Structure: Lectures: 3/Labs: 0
Credit Hours: 3
Prerequisites: None

Course Objectives:
This course covers a broad range of topics in computational linguistics/natural language processing, including word and sentence tokenization, text classification and sentiment analysis, spelling correction, information extraction, parsing, meaning extraction, and question answering. We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modeling, naïve bayes and maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning.

Course Syllabus:
Introduction, Chomsky hierarchy, Language models, Probability concepts, Bayes' Theorem, Smoothing

Course Outline:

1. Introduction, Why is NLP hard? Levels of language, NLP applications, Random language via n-grams. [J&M Ch. 1]
2. Chomsky hierarchy, What's wrong with n-grams? Regular expressions, CFGs, & more, Lists, trees, and vectors. [J&M Ch. 16]
4. Probability concepts, Joint & conditional prob, Chain rule and backoff, Modeling sequences, Cross-entropy and perplexity. [M&S 2]
5. Bayes' Theorem, Smoothing n-grams, Maximum likelihood estimation, Bias and variance, Add-one or add-lambda smoothing, Good-Turing discounting, Smoothing with backoff, Deleted interpolation, Conditional log-linear models (interactive visualization), Regularization. [M&S 2, 6; J&M 4]
8. Earley's algorithm, Top-down parsing, Earley's algorithm; Extending CFG, CCG, TSG, TAG [J & M 14]
9. Probabilistic parsing, Dependency grammar, Lexicalized PCFGs; Parsing tricks, Pruning; best-first, Rules as regexps, Left-corner strategy, Smoothing, Evaluation, A song about parsing; Human sentence processing, Methodology, Frequency sensitivity, Incremental interpretation, Unscrambling text; [M&S 12; J&M 14; TT]
10. Semantics, What is understanding? Lambda terms, Semantic phenomena and representations, more semantic phenomena and representations, Adding semantics to CFG rules, Compositional semantics. [J&M 17-18;]
11. Forward-backward algorithm, Ice cream, weather, words and tags, Forward and backward probabilities, Inferring hidden states, Controlling the smoothing effect, Reestimation, Likelihood convergence, Symmetry breaking, Local maxima, Uses of states. [J&M 6]
12. Expectation Maximization, Generalizing the forward-backward strategy, Inside-outside algorithm, Posterior decoding; Finite-state algebra, Regexp review, Properties, Functions, relations, composition, Simple applications; [M&S 11; R&S 1]
14. Programming with regexps, Analogy to programming, Extended finite-state operators, Date parsing, FASTUS; Morphology and phonology, English, Urdu, Arabic, Stemming, Compounds, segmentation, Two-level morphology, Punctuation, Rewrite rules, OT; Optimal paths in graphs, The Dyna perspective; [R&S 2]

15. Structured prediction, Perceptrons, CRFs, Feature engineering, Generative vs. discriminative; Current NLP tasks and competitions, The NLP research community, Text annotation tasks, Other types of tasks; Applied NLP, Topic models (Guest lecture); Machine translation (Guest lecture). [J&M 25, M&S 13]

Reference Material:

- Steven Bird. 2009. Natural Language Processing with Python. O'Reilly. (Free on SearchWorks)
- Psycholinguistics: Tanenhaus & Trueswell (2006), Human Sentence Processing website

Course Title: Advance Natural Language Engineering
Course Code: CS-8840
Course Structure: Lectures: 3/Labs: 0
Credit Hours: 3
Prerequisites: None

Course Objectives:
In this course we will explore statistical, model-based approaches to natural language processing. There will be a focus on corpus-driven methods that make use of supervised and unsupervised machine learning approaches and algorithms. We will examine some of the core tasks in natural language processing, starting with simple word-based models for text classification and building up to rich, structured models for syntactic parsing and machine translation. In each case we will discuss recent research progress in the area and how to design efficient systems for practical user applications.

Course Syllabus:
Course Outline:

1. Introduction & Language Modeling, MT Tutorial, Chen & Goodman, Kneyser-Ney. Large Language Models. [J & M Ch. 1, 4; or M & S Ch. 1-3, 6]
2. Text Classification via Naïve Bayes & Maximum Entropy, Classification Tutorial, MaxEnt Tutorial, Generative and Discriminative Classifiers. [J & M Ch. 6.6, 19.1, 20; or M & S Ch. 7]
3. Hidden Markov Models [J & M Ch.9]
4. Clustering, Part-of-Speech Tagging, TnT Tagger, Toutanova & Manning '00. [J & M Ch. 5; or M & S Ch. 10]
5. Advanced Part-of-Speech Tagging, Merialdo '94, CRFs, Johnson '07, Bilingual POS Induction. [J & M Ch. 6; or M & S Ch. 9]
6. Word Alignments, MT Tutorial, Overview, IBM Models, HMM-Alignments, Agreement. [J & M Ch. 25; or M & S Ch. 13]
7. Phrase-Based Translation, Decoding, Phrases, Moses [M & S Ch. 5,6]
8. Syntactic Parsing, Best-First, A*, K-Best, Shift-Reduce Parsing. [J & M Ch. 12,13; or M & S Ch. 3,11]
9. Advanced Constituency Parsing, Unlexicalized, Lexicalized, Latent Variable. [J & M Ch. 14; or M & S Ch. 12]
10. Semantic Parsing, Hierarchical (Syntax-Based) Translation, Hiero, GHKM, Syntax vs. Phrases, Synchronous Grammars. [J & M Ch. 23.3]
11. Sentiment Analysis, Aspects, Lexicons, Summarization. [J & M Ch. 23.4]
12. Summarization, Query, N-Gram, Topical. [B & L, pp. 1-135]
13. Lexical Acquisition [J & M Ch. 8]

Reference Material:


Course Title: Lexical Functional Grammar Development
Course Code: CS-7841
Course Structure: Lectures: 3/Labs: 0
Credit Hours: 3
Prerequisites: None

Course Objectives:
Natural language processing (NLP) enables computers to make use of data represented in human language (including the vast quantities of data available on the web) and to interact with computers on human terms. Applications from machine translation to speech recognition and web-based information retrieval demand both precision and robustness from NLP technology. Meetings these demands will
require better hand-built grammars of human languages combined with sophisticated statistical processing methods. This class focuses on the implementation of linguistic grammars, drawing on a combination of sound grammatical theory and engineering skills.

**Course Syllabus:**
Introduction, LFG, Templates, C & F description, Agreement, Determiners, Rules & alternations, Adjuncts, Obliques, Prepositions, Pronouns, Punctuation, Generation & Optimality, Complements, Uncertainty, Imperatives, Finite-State Morpholog, Free Word Order and the Shuffle Operator, Coordination

**Course Outline:**
1. Introduction and Overview, LFG Basics [M. D. 2001]
2. LFG Basics II, Templates I, MacOSX, Unix. [M. D. 2001]
3. Templates II, f-descriptions, Subject-Verb Agreement, Determiners, xlerc file [Crouch, 2008]
4. Lexical Rules, Passive and Argument alternations. [Crouch, 2008]
5. Adjuncts (Adjectives and Adverbs) and Obliques: PPs, Semantic and Non-Semantic Prepositions. [Crouch, 2008 & Butt, 1999]
6. Pronouns, Lexical Entries, Punctuation, Note on Adjuncts: Sets and Scope. [Crouch 2008]
7. Generation & Optimality Projection, Restricting Over-generation [Crouch 2008]
8. Complements, xcomp and comp. [Crouch 2008]
9. Functional Uncertainty, Imperatives and empty categories. [Crouch 2008]
10. Finite-State Morphology (FSM) I. [Crouch 2008]
11. FSM II (-unknown), Free Word Order and the Shuffle Operator. [Crouch 2008]
12. Meta-categories, Metarulemacros and Coordination. [Crouch 2008]
13. Project

**Reference Material:**