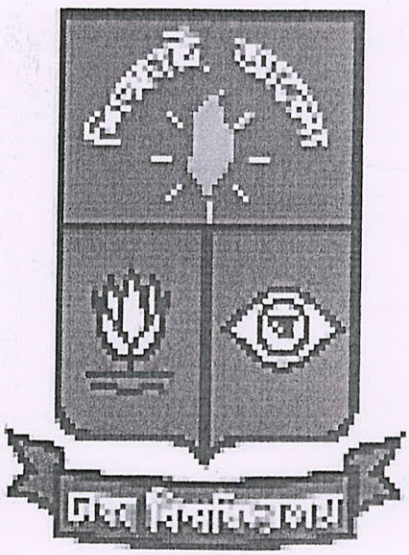
Proposal for the Master of Science in Software Engineering (MSSE) program



Institute of Information Technology

University of Dhaka

Prepared By –

MSSE Syllabus Committee, Institute of Information Technology, University of Dhaka.

- **Dr. Kazi Muheymin-Us-Sakib (Co-ordinator)**
  PhD (Melbourne, Australia), M.Sc (CSE, DU), B.Sc (CSE, DU)
  Email: sakib@univdhaka.edu

- **Professor Zerina Begum**
  PhD (DU), MS (PHY, DU), B.Sc (PHY, DU)
  Email: zerina@univdhaka.edu

- **Rezvi Shahriar**
  MS (CSE, DU), B.Sc (CSE, DU)
  Email: rezvi@univdhaka.edu

- **Dr. Md. Shariful Islam (Member Secretary)**
  PhD (Korea), MS (Sweden), B.Sc (CSE, DU)
  Email: shariful@univdhaka.edu

- **Advisors:**
  - **Dr. Mahbubul Alam Joarder**
    Director,
    Institute of Information Technology, University of Dhaka.

  - **Zulfiquar Hafiz**
    Associate Professor,
    Institute of Information Technology, University of Dhaka.

  - **Shafiul Alam Khan**
    Assistant Professor,
    Institute of Information Technology, University of Dhaka.

  - **Moniruzzaman Bhuiyan**
    Assistant Professor,
    Institute of Information Technology, University of Dhaka.

  - **Ahmedul Kabir**
    Lecturer,
    Institute of Information Technology, University of Dhaka.

  - **Nowshin Nowar**
    Lecturer,
    Institute of Information Technology, University of Dhaka.

  - **Sumon Ahmed**
    Lecturer,
    Institute of Information Technology, University of Dhaka.

  - **Nurul Ahad Tawhid**
    Lecturer,
    Institute of Information Technology, University of Dhaka.

  - **Emon Kumar Dey**
    Lecturer,
    Institute of Information Technology, University of Dhaka.

Table of Contents

1. Institute of Information Technology (IIT)

In the annual session of June 2001, The Dhaka University Senate established the Institute of Information Technology (IIT). Since then, IIT has produced numerous efficient manpower in information technology and has provided network services to the University. IIT also has put its foot marks on government and non-government sectors by providing consultancy services and various research and development work.

The institute currently offers degrees in three fields. Bachelor of Information Technology major in Software Engineering (BIT-SE) has become it's flagship by attracting meritorious students for doing their undergrad. Masters in Information Technology (MIT) is for pro-industry graduate program for IT related students and Post Graduate Diploma in Information Technology (PGDIT) gives opportunity to non-IT students to build their career in Information Technology. In addition, various short course programs related to IT are also offered here all over the year for all kind of pupils. IIT offered programs and student capacity is given in Table: 1.

	Program Name	No of Students/Ac Year
1	Bachelor of Information Technology Major is Software Engineering (BIT-SE) (4 Year)	140
2	Masters in Information Technology (MIT)	35
3	Post Graduate Diploma in Information Technology (PGDIT)	30

Table 1: IIT offered programs and enrolled students per Academic year

It is noteworthy that IIT has started it's Bachelor program from year 2009 and it's first honors batch is about to finish their bachelor program. IIT does not have any regular research based Masters program to accommodate those prospective students. Moreover, it is the demand from the software industry of Bangladesh, that IIT should start the software engineering research, as it did by opening the Bachelor degree in software engineering first ever in this country. To fulfill the software industry demand, IIT is now planning to offer a research based Masters program in software engineering, that is, **Master of Science in Software Engineering** (MSSE).

Research is also another important and inseparable part of IIT. At this moment, It has sixteen permanent faculty positions who are doing research in Software Engineering, Software security, Software risk assessment and analysis, Data acquisition system, Semiconductor technology, Reversible Logic Design, SOCs, Development of computerized systems, Computational fluid dynamics, Numerical analysis, Network design and deployment, Constraint satisfying problem, Adaptive web browser, Wireless communications, Value added systems for mobile, Authentication technology, File system simulation, Secure software design, IT risk management,

Network security, Sensor & Ad-hoc networks, Data Mining, Machine Learning, Database Systems, Bioinformatics and Computational Biology. This capacity gives IIT the strength to start a new endeavor in research based Masters program.

Students are the central point of any educational program. IIT can have the privilege to mention that last four years it is getting the top ranked students from the very competitive KA-UNIT Dhaka University admission test. Including the all science departments and institutes, IIT has been ranked third, based on the prospective student's choice. Excluding the biological departments or institutes, IIT is the first choice among the undergraduate prospective students for the fourth consecutive years (2009-2012). These students are another strength of IIT to offer them a new challenge of Masters program in Software Engineering.

Infrastructure is the last point that should be ready for offering any new academic program. IIT has a salient ambiance and is adjacent to the Dhaka University Science Library. The four-storied red ceramic IIT building contains six microcomputer laboratories, one microprocessor laboratory and two large classrooms. The laboratories and classrooms are air-conditioned and equipped with multimedia, document camera system and Internet facilities. The infrastructure capacity that IIT has is sufficient to start the proposed Master of Science in Software Engineering program.

2. Master of Science in Software Engineering (MSSE)

Software engineering is a systematic and disciplined approach to developing software. It applies computer science, mathematics and business principles and practices to the creation, operation and maintenance of software systems. It is now more than an integrated approach of design, development, QA and maintenance rather than only programing. Therefore it becomes increasingly important that while understanding developing technologies and architectures and their influence on software engineering processes, research for novel methods and methodologies on those very issues. The research is the only sustainable way of meeting the ever increasing global demands.

CNN/Money Magazine (2011) just rated software engineering as the number one best job for salary and opportunities *[http://money.cnn.com/magazines/moneymag/best-jobs/]*. Software Engineering graduates gain employment as software developers and testers, software architects and designers, team leaders and project managers, and executive-level positions in software development projects. At the same time, the appeal of research and development on new findings gradually creating inroads in this popular market. Even in Bangladesh, world reputed companies like Samsung, LG and many others are opening their research and development centers.

The proposed Master of Science in Software engineering degree will encompass a wide range of activities covering software engineering practice, software architectures, formal design methods, automated verification, computational models and foundations in concord with the traditional computer sciences. Mary Shaw shows in his famous journal on research in software engineering that there are six typical phases in this very new area of research which distinguished software engineering

research from other traditional fields like Physics, Chemistry etc. Those are: Basic research, Concept formulation, Development and extension, Internal extension and exploration, External extension and exploration and Popularization. The aim of this proposed program is to introduce those formal phases to the prospective students, and thus to make them ready for this newest challenge in the industry.

This will be a three semester (one and half year) full time program, where there will be two explicit parts. Half the semester credits will be allocated for theoretical studies to prepare students to do research. The rest will be allocated to do research and formal publications.

This will be an elite degree for those students who have excellent grades and also possess strong communication skills together with an outward worldly focus and personality.

2.1 Academic Requirements and Regulations at a glance

- The minimum duration of the MSSE course shall normally be three semesters. A candidate for the Master's degree must complete all the requirements for the degree within three academic years (session) from the date of the first admission in the respective program.

- Academic progress shall be measured in terms of credit hours earned by a student. One credit hour subject shall normally require one hour of class attendance per week for one semester; while one credit hour of thesis/project should normally require two hours of work per week for one semester including guided and unguided supervision; also one credit hour of laboratory should normally require two hours of work per week for one semester. The number of credit hours for each subject shall be as specified in the syllabus.

- For the degree of MSSE a student must earn a minimum of 36 credit hours including a thesis for which 18 credit hours shall be assigned.

- Since this is a regular program, all the students shall be considered as regular students and shall be expected to finish within the stipulated time of three semesters.

- According to the rule of the University of Dhaka, the registration of a Masters student is valid for three years. This rule imposes the constraints of finishing the degree within the registration period. In case of failure of finishing the degree within the registration period shall force the student to re-registrar and start the degree as a fresher.

- Regular students must enroll for 12 credit hours per semester. A regular student shall not be allowed to be in the employment of any organization (even as a part time employee). However, they may be employed as teaching / research assistant at the University. If a regular student becomes an employee (full-time or part-time) of any other organization in the middle of a semester,

s/he may, with the approval of the Head of the Institute and her/his employer, be allowed to continue as a regular student for that semester.

2.2 Entry Requirements

- The seats for MSSE program are limited.
- Candidates having good grades and have completed their undergraduate degree from IIT through Bachelor of Information Technology major in Software Engineering (BIT-SE) (internal candidates), shall get admission to the Master of Science in Software Engineering (MSSE) automatically.
- In case of lower number of Internal candidates than the MSSE program capacity, External candidates may be selected for those available seats.
- External Candidates (that is candidates from outside the IIT) must have fulfill the following criteria:
    - The age of the candidate must not be more than 30 at the time of application.
    - The candidate must have an undergraduate degree in Software Engineering or Computer Science.
    - The candidate must have CGPA not less than 3.5 in his/her undergraduate degree.

All the prospective candidates must sit for an admission test conducted by the IIT authority, and IIT shall have the ultimate authority to select students from that test subject to availability of seats.

2.3 Program Structure

The Master of Science in Software Engineering degree will provide a systematic, work-integrated study spanning the development, maintenance and operation of software. The degree will be particularly suited to people with strong communication skills and who will work at the interface between a software system and a business organization. The program structure is elaborated in Table 2.

Semester	Courses (Credits)			
1	Research Methodology (3)	Formal methods and Models in Software Engineering (3)	Secure Software Design and Programming (3)	Distributed Software Engineering (3)
2	Elective 1 (3)	Elective 2 (3)	Thesis (6)	
3	Thesis (12)			

Table 2: MSSE Program Structure

2.3.1 Credit distribution

In order to understand the program structure, the semester wise credit distribution is shown in Table 3.

Semester	Course Code	Course Title	Credit	Theory	Lab	Total Hours/ week
1	**MS1001**	Research Methodology	3	2	1	4
	MS1002	Formal methods and Models in Software Engineering	3	2	1	4
	MS1003	Secure Software Design and Programming	3	2	1	4
	MS1004	Distributed Software Engineering	3	2	1	4
2	**MS1XXX**	Elective 1	3	2	1	4
	MS1XXX	Elective 2	3	2	1	4
		Thesis	**Credit**	**Unguided Supervision**	**Guided Supervision**	**Total Hours/ week**
	MS1007	*Thesis	6	11	1	12
3	**MS1007**	*Thesis	12	22	2	24

Table 3: Course and credit distribution according to the semesters.

** Thesis (MS1007) shall be considered as a single course of 18 credits, however spanned over 2nd and 3rd Semester. The evaluation of this course will be done at the end of 3rd semester.*

2.3.2 Elective Courses

Elective courses (6 credits): A student shall take 2 elective courses from the courses available in the following emphasize fields. It is noteworthy that elective course list that is provided below is not exclusive. Faculties having expertise with new technologies can design and offer new elective courses.

a. Software design:

1. User Interface design and development
2. Reusable software architecture
3. Quality of services for software architectures

b. Software Assurance:

4. Information security theory and practices
5. Software engineering and security architecture
6. Security risk analysis and management

c. Software management:

7. Software engineering economics
8. Software project management
9. Security risk analysis and management

d. Applications:

Area1: Distributed Systems and networking

10. Cloud computing
11. Wireless mesh networks
12. Mobile and wireless security
13. Embedded systems

Area 2: Information retrieval and database systems

14. Distributed databases
15. Data warehousing and mining

Miscellaneous fields:

16. Bio-informatics and systems biology
17. Computer graphics and image processing
18. Game theory
19. Simulation and modeling
20. Probability and stochastic process
21. Algorithm and optimization
22. Artificial Intelligence and Machine learning

2.4 Course Details

Course listings with their course codes are given in Table 4 and 5. The course details that will be covered during the MSSE program are attached in Appendix A.

Number	Course Code	Course Name
1	**MS1001**	**Research Methodology**
2	**MS1002**	**Formal methods and Models in Software Engineering**
3	**MS1003**	**Secure Software Design and Programming**
4	**MS1004**	**Distributed Software Engineering**
5	**MS1XXX**	**Elective**
6	**MS1XXX**	**Elective**
7	**MS1007**	**Thesis**

Table 4: Course list

Number	Course Code	Course Name
8	**MS1008**	User Interface design and development
9	**MS1009**	Reusable software architecture
10	**MS1010**	Quality of services for software architectures
11	**MS1011**	Information security theory and practices
12	**MS1012**	Software engineering and security architecture
13	**MS1013**	Security risk analysis and management
14	**MS1014**	Software engineering economics
15	**MS1015**	Software project management
16	**MS1016**	Cloud computing
17	**MS1017**	Wireless mesh networks
18	**MS1018**	Mobile and wireless security
19	**MS1019**	Embedded systems
20	**MS1020**	Distributed databases
21	**MS1021**	Data warehousing and mining
22	**MS1022**	Artificial Intelligence and Machine learning
23	**MS1023**	Bio-informatics and systems biology
24	**MS1024**	Computer graphics and image processing
25	**MS1025**	Game theory
26	**MS1026**	Simulation and modeling

27	**MS1027**	Probability and stochastic process
28	**MS1028**	Algorithm and optimization

Table 5: Elective Course list

It is suggested that the course details should be reviewed in every three years time. The main difficulty of teaching technological subjects is their continuous evolution. To chase the cutting edge, technological subjects should be reviewed in a short period of time.

2.5 Evaluation Procedure

2.5.1 Guidelines for Letter Grading System

The Master of Science in Software Engineering (MSSE) degree program in the Institute of Information Technology, University of Dhaka is an one and half year program consisting of **three semesters** having duration of **six months each**, and consists of:

15 weeks	for holding classes
1 week	preparation time for examinations
3 weeks	semester final examination
3 weeks	result publication
4 weeks	vacation and holidays

2.5.1.1 Definition of a Credit

The Credit is defined as follows (also shown in Table 6):

i. Most of the courses will consist of both theoretical classes and laboratory works.
ii. The total number of credits of a course will be distributed for both theoretical classes and laboratory works as follows:

Class	Hours/week
1 Credit Theory	1
1 Credit Laboratory	2

Table 6: Credit definition

Marks	Letter Grade	Numeric Grade	Comments
80% or above	A+	4.00	Excellent
>=75% but < 80%	A	3.75	Better
>=70% but < 75%	A-	3.50	Good
>=65% but< 70%	B+	3.25	Above average
>=60% but <65%	B	3.00	Average
>=55% but <60%	B-	2.75	Below average
>=50% but <55%	C+	2.50	Satisfactory
>=45% but <50%	C	2.25	Not satisfactory
>=40% but <45%	D	2.00	Pass
Less than 40%	F	0.00	Fail

Table 7: Grading system

2.5.1.2 Letter Grade, Grade Points and their Meaning

Grades in each course will be assigned (in accordance with the rules of Faculty of Science, DU) as mentioned in Table 7.

The GPA (Grade Point Average) and CGPA (Cumulative Grade Point Average) will be calculated as follows:

$$\text{GPA or CGPA} = \frac{\text{Grade Points in a Course X credits for that course}}{\text{Total Credits taken}}$$

2.5.2 Promotion to the next Semester

i. To be promoted to the next semester, the overall CGPA obtained by a student in the previous semester must not be less than **2.5** and he/she have to obtain **at least grade D** individually in each course.
ii. Student who will get fail (in case of grade **F**) in a particular course will sit for a **supplementary exam** for it and he /she must obtain at least grade **D** in that course.
ii. Students who get **F** in a particular course will retake it with the next batch.

2.5.3 Final Exam Requirements

i. To sit for the final exam of a particular course a student must have total attendance **not less than 60%.**
ii. The overall attendance **to be collegiate must not be less than 75%.** Only collegiate students will take part in the semester final exams. However, a student whose overall attendance is **equal or above 60% but less than 75% will be subject to financial penalty**.

2.5.4 Assessment and Evaluation

Except the thesis, the total performance of a student in a given course will be based on continuous assessment and course final examinations. Marks distribution will be as follows:

Continuous Assessment	**50% of total marks**
Course Final Examinations	**50% of total marks**

The continuous assessment may consist of class tests, attendance, seminars/presentations/viva-voce, assignments, completion of projects, and mid term examinations. The mentioned criteria to assess a student will be justified by individual course teacher and he/she may set his own assessment criteria **except final examinations**.

Assessment of Thesis Course:

All the students have to submit a thesis book at the end of their third semester. The thesis will be evaluated by two examiners: two experts of the area of research (experts can be selected from IIT or from outside of the IIT selected by the supervisor of that thesis). The evaluation of the thesis will be as follows:

- Pass with no corrections
- Pass with minor corrections
- Pass with major corrections
- Fail

After having experts' evaluation report, a student may have to revise his/her thesis and resubmit the thesis. In case of having a fail grade, a student has to retake the whole 18 credits of thesis.

After completion of writing the thesis, with the consent of the supervisor, a student may apply for his/her thesis defense. The supervisor will arrange a rigorous thesis presentation at the seminar room (open to all) and then arrange an interview session with the higher research degree committee of IIT and some experts on that area of research. This presentation and interview will carry 50% of the total marks and the supervisor will provide the other 50% through continuous evaluation.

2.5.5 Credit Requirements and Passing Criteria

i. Completion of minimum 36 credits.
ii. Passing of all courses individually with at least D grade.
iii. Grade point average (GPA) of 2.5 or above.

2.6 Program Fee Structure

Since this will be the regular MSC program offered within the jurisdiction of The University of Dhaka, the program fee structure will be as per university rule. However, if any additional costs should incur, IIT holds the right to charge those extra fees with prior permission from the Academic Committee of IIT.

3. Final Remark

Software Engineering is an important stream of typical computer science what IIT is planning to address. There are other streams such as, database, networking etc, which have acute market demands. The future goal of IIT is set to produce quality researchers in those streams as well.

IIT authority will resolve any other points not mentioned in this document.

APPENDIX A

Core courses (12 credits): These are the courses that are mandatory for all the students to do research in Software Engineering. Usually in the first semester of MSSE, students shall have to take all these courses.

1. MS1001 (Research methodology):

This course intends to prepare graduate students for advanced research in Software Engineering by studying how to plan, conduct and report on empirical investigations.

Course contents:

Research methods and techniques: Software engineering data collection for field studies, qualitative methods, simulation methods

Practical foundations: Statistical methods and measurements, Missing data in software engineering, reporting experiments in software engineering, A practical guide Ethical research involving humans

Knowledge creation: Selecting empirical methods for software engineering research, building theories in software engineering

Paper writing and publication: Introduction to latex, paper organization and presentation

Suggested Readings:

- Guide to Advanced Empirical Software Engineering Forest Shull, Janice Singer and Dag Sjoberg (Eds.), Springer Verlag, 2008
- Horn, R., Schwarzkopf, A. and Price, R., "Information Systems Solutions: A project approach". McGraw Hill, 2006
- Holtom, Daniel. "Enjoy writing your science thesis or dissertation! : a step by step guide to planning and writing dissertations and theses for undergraduate and graduate science students", Imperial College Press. 1999
- Robert A Day. "How to write and publish a scientific paper". Oryx Press, 1989.
- P D Leedy. "Practical Research: Planning and Design", Collier Macmillan, 1989.
- Lindsay D. "A Guide to Scientific Writing", Longman, 1995

2. MS1002 (Formal Methods and Models in Software Engineering):

The objective of this course is to introduce the Formal mechanisms for specifying, validating, and verifying software systems, integration of formal methods with existing programming languages, and the application of formal methods to requirements analysis, testing, safety analysis, and object-oriented approaches. Formal methods using the Object Constraint Language (OCL).

Course contents:

Introduction: Formal methods, developing and acquiring formal methods, applying formal methods, logic and theorem proving

Modeling software system: Sequential, concurrent and reactive systems, transition systems, example of modeling programs

Formal specification: Properties of specification formalism, Linear temporal logic, Automata on Infinite words, complicated specifications

Automatic verification: State space search, representing states, the automata framework, advantages and weaknesses of model checking

Deductive Software Verification: Verification of flowchart programs, verification with array variables, axiomatic program verification, advantages and weaknesses of deductive verification.

Suggested Readings:

- Diller, Z An Introduction to Formal Methods (2nd ed.), Wiley, 1994.
- M. Huth and M. Ryan. Logic in Computer Science: Modeling and reasoning about systems. Cambridge University Press. 2nd Edition. 2004.
- D. Peled.Software Reliability Methods. Springer Verlag. 2001.
- Formal Methods and Software Engineering, 7th International Conference on Formal Engineering Methods, ICFEM 2005, Manchester, UK, November 1 – 4, 2005.
- J. Warmer and A. Kleppe. The Object Constraint Language. Precise Modeling with UML. Addison Wesley. 1999.

3. **MS1003 (Secure Software Design and Programming):**

This course intends to provide Theory and practices of software security with a view to design software with security in mind from the ground up and to integrate analysis and risk management throughout the software life cycle.

Course contents:

Introduction: Security principles, concept of computer security, security services and policies

Security risks: Database security, operating systems security, secure coding

Countermeasures: methodologies and tools for identifying and eliminating security vulnerabilities, techniques to prove the absence of vulnerabilities, and ways to avoid security holes in new software.

Secure software design: essential guidelines for building secure software, information security standards

Suggested Readings:

- Security in Computing, 4th Edition, by Charles P. Pfleeger , Publisher: Prentice Hall; 4th edition
- Computer security: principles and practices, by William Stallings and Lawrie Brown, 2nd Edition,
- Brian Chess and Jacob West, Secure Programming with Static Analysis (required)
- David A. Wheeler, Secure Programming for Linux and Unix HOWTO Version 3.5, Aug 2004 (required)
- Goertzel et al, Software Security Assurance State of the Art Report, May 2007.
- Aleph One, Smashing the Stack for Fun and Profit. Phrack Vol 7, Nr. 49
- Tim Newsham, Format String Attacks, Guardent tech report, Sept 2000

4. MS1004 (Distributed Software Engineering):

Hands-on introduction to techniques and programming interfaces for distributed software engineering. Networking protocols at several layers. Construction of distributed and concurrent software using network protocol services. Applications of Internet and Web-based software.

Course contents:

Introduction: Definitions of a distributed system, typed of distributed system, system architectures, architectures versus middleware

Key concepts: Threads, virtualization, client, server, code migration, remote procedure call; names, identifiers and addresses; synchronization, consistency and replication; fault tolerance and replication.

Distributed object based system: Architecture, process, communication, naming, synchronization, consistency and replication, fault-tolerance and security.

Distributed Web-Based Systems: Architecture, process, communication, naming, synchronization, consistency and replication, fault-tolerance and security.

Distributed File Systems: Architecture, process, communication, naming, synchronization, consistency and replication, fault-tolerance and security.

Distributed Coordination-Based Systems: Architecture, process, communication, naming, synchronization, consistency and replication, fault-tolerance and security.

Suggested Readings:

- Distributed Systems: Principles and Paradigms (2nd edition), Andrew S. Tanenbaum, Maarten Van Steen, 2007, ISBN 0-13-239227-5 (required)
- J. Waldo, G. Wyant, A. Wollrath, and S. Kendall. "A Note on Distributed Computing."
- W. Emmerich. "Software Engineering and Middleware: A Roadmap."
- Debu Panda, "An Introduction to Service-Oriented Architecture from a Java Developer Perspective."
- A. Carzaniga, G. P. Picco, and G. Vigna. "Designing Distributed Applications with Mobile Code Paradigms."

Elective courses

A student will take 2 elective courses from the courses available in the following emphasized fields. Elective courses are generally offered to cope with the state of the art demands. Hence, the list mentioned below is only a guideline; courses outside the list may also be offered if required.

a. **Specialization Area- Software design:**

MS1008 (User Interface design and development):

This course is for students from a range of backgrounds studying classes in human-computer interaction, interaction design, web design, etc. A broad range of professionals and technology users will also find this useful, and so will graduate students who are moving into this area from related disciplines. This course is concerned with a broader scope of issues, topics, and paradigms than has traditionally been the scope of human-computer interaction (HCI). This reflects the exciting times we are living in, when there has never been a greater need for user interface designers and usability engineers to develop current and next-generation interactive technologies. To be successful they will need a mixed set of skills from psychology, human-computer interaction, web design, computer science, information systems, marketing, entertainment, and business.

This course entails creating user experiences that enhance and extend the way people work, communicate, and interact. Now that it is widely accepted that HCI has moved beyond designing computer systems for one user sitting in front of one machine to embrace new paradigms, we, likewise, have covered a wider range of issues. These include ubiquitous computing and pervasive computing that make use of wireless and collaborative technologies. We also have tried to make the book up-to-date with many examples from contemporary research.

Course contents:

What is UI design: Good and poor design, What to design, What is UI design? The makeup of interaction design? Working together as a multidisciplinary team, Interaction design in business, What is involved in the process of interaction design? The goals of interaction design, Usability goals, User experience goals

Understanding and conceptualizing UI: Understanding the problem space, Conceptual models based on activities, Conceptual models based on objects

Understanding users: What is cognition? Applying knowledge from the physical world **to** the digital world, Conceptual frameworks for cognition, Mental models Information processing.

Designing for collaboration and communication: Social mechanisms used in communication and collaboration, Conversational mechanism, Designing collaborative technologies to support conversation, Coordination mechanisms.

Understanding how interfaces affect users: What are affective aspects? Expressive interfaces, User frustration, Dealing with user frustration, The process of interaction design: What is interaction design about? Four basic activities of interaction design, Three key characteristics of the interaction design Identifying needs and establishing requirements: Why establish requirements? What are requirements? Different kinds of requirements, Data gathering Design, prototyping and construction: What is a prototype? Low-fidelity prototyping, High-fidelity prototyping, Compromises in prototyping, Construction: from design to implementation, Conceptual design: moving from requirements to first design, Three perspectives for developing a conceptual model, Expanding the conceptual model, Using scenarios in conceptual design, Using prototypes in conceptual design

User-centered approaches to design: Why is it important to involve users at all? Degrees of involvement, What is a user-centered approach? Understanding users' work: applying ethnography in design, Coherence, Contextual Design

Introducing evaluation What **to** evaluate, Why you need to evaluate, Hutchworld case study, An evaluation framework, Evaluation paradigms and techniques, Evaluation paradigms, D E C I D E: A framework to guide evaluation, Determine the goals

Observing users: Goals, questions and paradigms, Approaches to observation, How to observe, In controlled environments, Asking users and experts, Asking users: interviews, Developing questions and planning an interview, Unstructured interviews, Structured interviews.

Testing and modeling users: User testing, Testing MEDLINE, Doing user testing, Determine the goals and explore the questions, Choose the paradigm and techniques.

Design and evaluation in the real world: Designing mobile communicators, Nokia's approach to developing a communicator, Philip's approach to designing a communicator for children, Redesigning part of a large interactive phone-based response system.

Suggested Readings:

- Lumsden, j. (ed.) (2008) Handbook of research on user interface design and evaluation for mobile technology
- Albert, B., Tullis, T. and Tedesco, D. (2010) Beyond the Usability Lab: Conducting Large-scale Online User experience studies

- Brinck, T., Gergle, D. and Wood, S.D. (2002) Usability for the Web: Designing Web Sites that Work
- Perlman, G. (2009) User Interface Usability Evaluation with Web-Based Questionnaire
- Dix, A., Finley, J., Abowd, G. and Beale, R. (2003) Human-computer interaction., Upper Saddle River, NJ: Prentice-Hall.
- Helal, A., Mokhtari, M. and Abdulrazak, B. (2008) The Engineering Handbook of Smart Technology for Aging, Disability and Independence, 1st edition, USA: Wiley-Interscience.
- Muhlhauser, M. and Gurevych, I. (2007) Handbook of research on ubiquitous computing technology for real time enterprises /, London: Information Science Reference.
- Rubin, J.Z., Chisnell, D. and Spool, J.M. (2008) Handbook of usability testing: how to plan, design, and conduct effective tests, 2nd edition, Indianapolis: Wiley Publishing.

MS1009 (Reusable software architecture):

Course contents:

Overview of software reuse: Designing reusable software, reusable components, Reusable Software Architectures, Software libraries, generic architectures, software product lines, modeling single systems and product lines with UML, model driven architecture.

Object-Oriented Software Life Cycle for Software Product Lines: Object-Oriented Requirements Modeling; Object-Oriented Analysis Modeling, Object-Oriented Design Modeling, Incremental software construction and integration.

Requirements Modeling for Software Product Lines: The use case modeling approach for defining functional requirements. Kernel, optional, and alternative use cases and actors. Modeling variability with use case parameterization, variation points, and extension points.

Feature Modeling for Software Product Lines : Feature as a reusable requirement. Common, optional, default, and alternative features. Feature dependencies. Feature groups – mutually exclusive, one and only one, one or more of a group. Feature Modeling with UML. Modeling features with use cases; relationship between features and use cases. Feature conditions.

Analysis Modeling for Software Product Lines: Static modeling: objects, classes, and relationships. Object and class structuring; class categorization using stereotypes. Kernel, optional, and variant classes.

Dynamic interaction modeling for Software Product Lines : Developing object interaction models for kernel, option, and alternative use cases. Developing interaction models for different scenarios addressing use case variability. Kernel First Approach. Software Product Line Evolution Approach.

Statecharts for Software Product Lines : Kernel, optional and variant statecharts. Mutually exclusive variant statecharts and co-existing variant statecharts. High-level statechart to capture generalization of multiple variants. Modeling variability in statecharts through inheritance and parameterization. Feature/class dependency modeling. Modeling commonality/variability with abstract classes and parameterized classes. Using inheritance to support variant classes. Feature-based impact analysis. Feature/class dependencies.

Architectural Patterns for Software Product Lines: Software architectural structure patterns, software architectural communication patterns, software architectural transaction patterns, documenting software architectural patterns, applying software architectural patterns in product lines.

Software Component-based Architectural Design for Product Lines: Developing the overall software architecture. Separation of concerns in component design. Component-based structuring criteria. Software components and connectors: ports, provided and required interfaces.

Software Application Engineering: Deriving individual members of the product line from the OO software product line architecture and components. Using the product line feature model to derive the requirements, analysis, and design models for the application.

Software Product Line Case Studies : Microwave Oven, Distributed Factory Automation, Electronic Commerce.

Suggested Readings:

- Hassan Gomaa, "Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures", Addison-Wesley Object Technology Series, 2005.
- F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, "Pattern Oriented Software Architecture: A System of Patterns", John Wiley & Sons, 1996.

MS1010 (Quality of services for software architectures)

Course contents:

The Architecture Business Cycle : Where Do Architectures Come From?, Software Processes and the Architecture Business Cycle, What Makes a "Good" Architecture?

What Is Software Architecture? : What Software Architecture Is and What It Isn't, Other Points of View, Architectural Patterns, Reference Models, and Reference Architectures, Why Is Software Architecture Important?, Architectural Structures and Views.

A-7E Avionics System: A Case Study in Utilizing Architectural Structures Relationship to the Architecture Business Cycle, Requirements and Qualities, Architecture for the A-7E Avionics System.

Understanding Quality Attributes : Functionality and Architecture, Architecture and Quality Attributes, System Quality Attributes, Quality Attribute Scenarios in Practice, Other System Quality Attributes, Business Qualities, Architecture Qualities.

Achieving Qualities : Introducing Tactics, Availability Tactics, Modifiability Tactics, Performance Tactics, Security Tactics, Testability Tactics, Usability Tactics, Relationship of Tactics to Architectural Patterns, Architectural Patterns and Styles.

Air Traffic Control: A Case Study in Designing for High Availability Relationship to the Architecture Business Cycle, Requirements and Qualities, Architectural Solution.

Designing the Architecture : Architecture in the Life Cycle, Designing the Architecture, Forming the Team Structure, Creating a Skeletal System.

Flight Simulation: A Case Study in an Architecture for Integrability Relationship to the Architecture Business Cycle, Requirements and Qualities, Architectural Solution

Suggested Readings:

- Software Architecture in Practice, 2nd edition, Len Bass, Paul Clements, and Rick Kazman. Addison-Wesley, 2003
- Software Architecture: Foundations, Theory, and Practice . R. N. Taylor, N. Medvidovic, and E. M. Dashofy. Wiley, 2009.

b. **Specialization Area: Software Assurance:**

MS1011 (Information security theory and practices)

Course contents:

Introduction : An Overview of Computer Security
Foundations : Access Control Matrix, Foundational Results

Policy : Security Policies, Confidentiality Policies, Integrity Policies, Hybrid Policies, Noninterference and Policy Composition

Implementation I: Cryptography
Basic Cryptography, Key Management, Cipher Techniques, Authentication

Implementation II: Systems
Design Principles, Representing Identity, Access Control Mechanisms, Information Flow, Confinement Problem

Assurance (contributed by Elisabeth Sullivan)
Introduction to Assurance, Building Systems with Assurance, Formal Methods, Evaluating Systems

Special Topics Malicious Logic, Vulnerability Analysis, Auditing, Intrusion Detection

Practicum : Network Security, System Security, User Security, Program Security

End Matter : Lattices, The Extended Euclidean Algorithm, Entropy and Uncertainty, Virtual Machines, Symbolic Logic, Example Academic Security Policy, Bibliography

Suggested readings:

- Computer Security: Art and Science by Matt Bishop. Addison-Wesley Pub Co; 1st edition (December 2, 2002)
- Information Security : Principles and Practice by Mark Stamp Wiley-Inter-science; 1 edition (October 28, 2005)

MS1012 (Software engineering and security architecture):

The security of software systems in recent years has been transformed from a mono-dimensional technical challenge to a multi-dimensional techno-social challenge, due to the wide usage of software systems in almost every area of the human life. This situation requires a different and more holistic approach to the development of secure software systems.

Software Engineering & Secure Systems course may present the most recent and innovative lines of research and industrial practice related to secure software engineering. It may provide coverage of recent advances in the area of secure software engineering that address the various stages of the development process from requirements to design to testing to implementation. Contributions offer a comprehensive understanding secure software engineering, inspire and motivate further research and development, and bridge the gap between academic research and industrial practice.

Course contents:

Comparing modeling: approaches for security patterns, Incorporating social trust into design practices.
Model-based analysis of control systems: Network system configuration and management, Privacy aware systems.
Security and performance during system design: Security over the information system development cycle, State model diagrams,
State of practice in secure software: Using security patterns to develop secure systems

Suggested Readings

- Software Engineering for Secure Systems: Industrial and Research Perspectives by H. Mouratidis (University of East London, UK)
- Computer security: principles and practices, by William Stallings and Lawrie Brown, 2nd Edition

MS1013 (Security risk analysis and management):

Course contents:

Overview of Risk: Introduction to the terminology of risk; Risk analysis and management framework; Risk and the relationship to security and controls;

Assets, Harm & Threats: Types of harm; Impact valuation; Threat modelling;

Vulnerability Analysis: Baseline systems; Vulnerability analysis methods;

Risk Evaluation: Quantitative and Qualitative risk measurement;

Security Planning & Management: Security measures; Mapping vulnerabilities to mitigation techniques; COBIT; Security checklists; Security requirements; Risk standards;

Business Cases for Security: Building business cases; Saved losses;

Suggested readings:

- Information Security Risk Analysis, Thomas R. Peltier. Auerbach Publications, April 2005.
- Assessing and Managing Security Risk in IT Systems: A Structured Methodology,
- John McCumber, Auerbach Publications; June 2004
- *Managing Information Security Risks: The Octave Approach*
- Christopher J. Alberts, Audrey J. Dorofee, Addison-Wesley Professional; July 2002
- Probabilistic Risk Analysis: Foundations and Methods, T. Bedford, Roger M. Cooke, Cambridge University Press.

c. **Specialization Area- Software management:**

MS1014 (Software engineering economics):

Course Outline:

Programming aspects, economic aspects, human relations aspects, software trends: cost, social impact, the plurality of SE Means, The GOALS Approach to Software Engineering, The Software Work Breakdown Structure (WBS), Software Maintenance, introduction to COCOMO, definitions and assumptions, development effort and schedule, phase distribution, The Raylaigh Distribution, interpolation, basic software maintenance effort estimation. Performance Models, Optimal Performance, Sensitivity Analysis, Cost-Effectiveness Models.

Suggested readings:

- Boehm, Software Engineering Economics, Prentice Hall, 1981.
- Boehm et al., Software Cost Estimation with COCOMO II , Prentice Hall, 2000.
- Reifer, Don. Making the Software Business Case: Improvement by the Numbers, Addison Wesley, 2001.

MS1015 (Software project management)

This subject aims to present and develop the confidence and software project management skills required to become effective project team leaders and potential project managers. The course covers such concepts as team constitution, business aspects, technical organisations charts and cost estimates, scheduling and monitoring, and maintenance. The course proposes an analysis of existing software project management tools and groupware technologies. Apart from the theoretical presentations, much time is given to participants reviewing their past experience and doing illustrative exercises.

Course contents:

Role of the Project Manager: Outline, Staff, Introduction, What is a 'Project'?, Project? What Project?, Classical Engineering Projects and Soft ware Projects, Software Methodologies, Software Project Types, Tasks of a Software Project Manager, The Triple Constraint.

Project Startup: So The Project's Starting, Negotiating Project Constraints, Cost Benefit Analysis, Startup Deliverables, Other Considerations, System Requirements, The Role of System Requirements, Deliverables, Objectives of the Requirements Specification , Creating the Requirements Specification, Analysis Phase Activities

Methodologies: Methodology, Phases and Activities, Deliverables, Other Lifecycle Models, Estimation, Why Estimate?, When are Estimates Done?, Estimation Methods

Project Framework: Elements of the Framework, Standards and Quality, Framework Activities,

Project Monitoring: Purpose of Monitoring, Monitoring Expenditure, Monitoring Work, Cost/ schedule Milestones, Earned value, Managing *Deviations and Variations, Technical Audits, Project Progress Reporting,*

Human Factors: People in the Project, Staff as Individuals, Management, Managing Staff, Change Management, Improving Processes, Risk, The Nature of Risk, Risk Management Plan, Risk Identification, Risk Analysis, Risk Response Planning, Completion Activities

Project Completion and Implementation: Project Team Actions at Completion, What You Have to Do When You Thought You Were Done, Learning Lessons, Professional Ethic

Suggested readings:

- Marchewka, J. T., "Information Technology Project Management", 2nd Ed., J Wiley, 2006
- Sommerville, I, "Software Engineering 7", 7th Ed., Addison Wesley, 2004
- Hughes, B and Cotterell, M, "Software Project Management", 3rd Ed, Mc-Graw Hill, 2002
- Humphrey, W, "Managing the Software Process", SET Series in Software Engineering, Addison Wesley 1990
- Boehm, B, "Software Engineering Economics", Prentice Hall, 1981
- Blanchard, B. S and Fabrycky W.J, "Systems Engineering and Analysis", Prentice Hall, NJ, 1997
- Peterson, "Software Project Management", supplied electronically

d. **Specialization Area-Applications:**

MS1016 (Cloud computing):

Introduction: Its potential for lowering IT costs makes cloud computing a major force for both IT vendors and users; it is expected to gain momentum rapidly with the launch of Office Web Apps later this year. Because cloud computing involves various technologies, protocols, platforms, and infrastructure elements, this comprehensive reference is just what you need if you'll be using or implementing cloud computing. Cloud computing offers significant cost savings by eliminating upfront expenses for hardware and software; its growing popularity is expected to skyrocket when Microsoft introduces Office Web Apps.
This course helps define what cloud computing is and thoroughly explores the technologies, protocols, platforms and infrastructure that make it so desirable, Covers mobile cloud computing, a significant area due to ever-increasing cell phone and smartphone use, Focuses on the platforms and technologies essential to cloud computing. Anyone involved with planning, implementing, using, or maintaining a cloud computing project can rely on this course.

Course Contents:
Changing the Way We Deliver Services with Cloud Computing, Demystifying Cloud Computing, Enabling Business Innovation by using Cloud Computing, How Cloud Computing Will Help Your Business, Strategies for Moving to the Cloud, Identifying the Right Cloud Architecture for Your Business, Roadmap to Cloud Computing: The Planning Phase, Roadmap to Cloud Computing: The Implementation Phase, Maintaining a Cloud Environment: Governance, Growth and Security,Key Steps in Establishing Enterprise Cloud Computing Services

Suggested readings:

- Enterprise Cloud Computing: A Strategy Guide for Business and Technology Leaders by Peter Fingar , Andy Mulholland & Jon Pyke

MS1025(Game Theory):

Static games of complete information: Decision Theory, Strategic Game, Nash Equilibrium, Mixed Strategy Equilibrium, Applications.

Dynamic games with complete information: Extensive Games, Backward Induction, Extensive-form representation of Games, Sub-game Perfect Equilibrium, Repeated Games, Applications.

Static games of incomplete information: Bayesian Games, Bayesian Nash Equilibrium, Applications.

Dynamic games with incomplete information: Perfect Bayesian Equilibrium, Signaling Games, Applications.

Principal-Agents Problems : Adverse selection, Moral hazard, Signalling Games

Repeated Games : Representation of repeated games, Infinitely repeated games, Finitely repeated games.

Other topics: Asymmetric Information and Mechanism Design, Evolutionary Games, Applications.

Suggested readings:

- *An Introduction to Game Theory* Martin Osborne, Oxford University Press, 2004. ISBN 0-19-512895-8; ISBN-13: 978-0-19-512895-6.
- Fudenberg, D., Tirole, J. (1991). Game Theory, MIT Press, Cambridge, Massachussets.
- Osborne, M., Rubinstein, A. (1994). A Course in Game Theory, MIT Press, Cambridge, Massachussets.
- Mas-Collel, A., Whinston, M. D., Green, J. R. (1995). Microeconomic Theory, Oxford University Press, New York, Oxford. (Chapters 13, 14)

MS1026 (Simulation and Modeling):

Introduction to Simulation and Modeling: Simulation – introduction, appropriate and not appropriate, advantages and disadvantage, application areas, history of simulation software, an evaluation and selection technique for simulation software, general – purpose simulation packages. System and system environment, components of system, type of systems, model of a system, types of models and steps in simulation study.

Manual Simulation of Systems: Simulation of Queuing Systems such as single channel and multi channel queue, lead time demand, inventory system, reliability problem, time-shared computer model, job-shop model.

Discrete Event Formalisms: Concepts of discrete event simulation, model components, a discrete event system simulation, simulation world views or formalisms, simulation of single channel queue, multi channel queue, inventory system and dump truck problem using event scheduling approach.

Statistical Models in Simulation: Overview of probability and statistics, useful statistical model, discrete distribution, continuous distribution, empirical distribution and Poisson process.

Queuing Models: Characteristics of queueing systems, queueing notations, long run measures of performance of queueing systems, Steady state behavior of Markovian models (M/G/1, M/M/1, M/M/c) overview of finite capacity and finite calling population models, Network of Queues.

Random Number Generation: Properties of random numbers, generation of true and pseudo random numbers, techniques for generating random numbers, hypothesis testing, various tests for uniformity (Kolmogorov-Smirnov and chi-Square) and independence (runs, autocorrelation, gap, poker).

Random Variate Generation: Introduction, different techniques to generate random variate:- inverse transform technique, direct transformation technique, convolution method and acceptance rejection techniques.

Input Modeling: Introduction, steps to build a useful model of input data, data collection, identifying the distribution with data, parameter estimation, suggested estimators, goodness of fit tests, selection input model without data, covariance and correlation, multivariate and time series input models.

Verification and Validation of Simulation Model: Introduction, model building, verification of simulation models, calibration and validation of models:- validation process, face validity, validation of model, validating input-output transformation, ttest, power of test, input output validation using historical data and Turing test.

Output Analysis: Types of simulations with respect to output analysis, stochastic nature of output data, measure of performance and their estimation, output analysis of terminating simulators, output analysis for steady state simulation.

Case Studies: Simulation of manufacturing systems, Simulation of Material Handling system, Simulation of computer systems, Simulation of super market, Cobweb model, and any service sectors.

Suggested Readings:

- Banks J., Carson J. S., Nelson B. L., and Nicol D. M., "Discrete Event System Simulation", 3rd edition, Pearson Education, 2001

- Bernard P. Zeigler, Herbert Praehofer and Tag Gon Kim, "Theory of Modeling and Simulation", Second Edition.

MS1027(Probability and Stochastic Process)

Probability space, Random variables and random vectors. Sample Space, Probabilities and Bayes' Formula, Random Variables, Expectation of a Random Variable and Jointly Distributed Random Variables, Conditional

Probability and Conditional Expectation, expectation. Inequalities. Characteristic function. Convergence of random sequences, Types of convergences. Law of large numbers. Central limit theorem. Discussion of the exponential distribution and the Poisson process. General definition of counting processes. Generalizations of the Poisson process. Renewal theory and its application. Queuing theory. Reliability theory and its applications. Brownian motion and stationary processes.

Stochastic processes: Classification of stochastic processes. Wide sense stationarity. Point processes. Poisson processes. Markov chains. Linear transformations of stationary processes. Doob decomposition. Stochastic Karhunen-Loeve expansions. Campbell Theorem.

Statistical signal processing: Statistical signal processing. Kalman filter. Wiener filter for random sequences. Stochastic simulation.

Suggested readings:

- Sheldon M. Ross (2010). Introduction to Probability Models, 10th ed. AP, ISBN-978-0-12-375686-2
- Billingsley, P. Probability and Measure. 3rd edition. ISBN: 0471007102, Wiley (1995)
- Kay, S. Intuitive Probability and Random Processes using MATLAB. ISBN: 0387241574, Springer (2005)
- Montes, F. Introducción a la Probabilidad. Notas de clase. Dpt. d'Estadistica i I. O.(2004)
-
- Balakrishnan, A. V. Introduction to Random Processes in Engineering. ISBN: 978- 0-471-74502-0, Wiley (2005)
- Billingsley, P. Probability and Measure. 3rd ·dition. ISBN: 0471007102, Wiley (1995)
- Dougherty, E. R. Random Processes for Image Signal Processing. ISBN: 0780334957, Wiley-IEEE Press (1998)

MS1028 (Algorithm and Optimization)

Course Contents:

Optimization methods: Introduction to methods of optimization; optimality and convexity, General optimization algorithm; necessary and sufficient conditions for optimality, single objective functions, multiobjective functions, constraint handling, unifying approaches.

Linear programming : Introduction to linear programming—a geometric perspective, Standard form in linear programming; basic solutions; fundamental theorem of linear programming, Simplex method; multiple solutions; tie-breaking procedures; two-phase method, Duality theory in linear programming; complementary slackness; economic interpretation of the dual, Sensitivity analysis; right-hand-side and cost ranging

Network flows: Minimum cost network flow algorithms; push relabel algorithms, out-of-kilter method; primal-dual methods

Integer programming: Introduction to Integer programming; applications in optimal scheduling in software engineering

Interior point methods: Interior point optimization methods; affine scaling method

Quadratic programming: Karush-Kuhn-Tucker conditions for constrained nonlinear programming problems; necessary and sufficient conditions; quadratic programming; applications

Machine learning algorithms: Heuristic optimization methods, Evolutionary algorithms: Areas of applications, Fitness assignment, Weighted sum fitness assignment, Sharing functions; genetic algorithm and genetic programming ; simulated annealing methods

Suggested readings:

- Linear and Nonlinear Optimization, by I. Griva, S. Nash, and A. Sofer, 2nd Edition,
- Society for Industrial and Applied Mathematics, 2009. [ISBN: 978-0-898716-61-0]
- Global Optimization Algorithms Theory and Application, Thomas Weise.
- Models and Algorithms for Global Optimization, by Amio Torn and Julius Iiinskas, Springer, November 2010.
- Stochastic Global Optimization and Its Applications with Fuzzy Adaptive Simulated Annealing, by Hime Aguiar e Oliveira Junior, Lester Ingber, Antonio Petraglia, Mariane Rembold Petraglia, Maria Augusta Soares Machado, Springer, January 2012.