CENTENNIAL COLLEGE

Course Outline

School:	Eng. Tech. & Applied Science
Department:	Information and Communication Engineering Technology (ICET)
Course Title:	Software Engineering Fundament
Course Code:	COMP 120
Course Hours/Credits:	56
Prerequisites:	N/A
Co-requisites:	N/A
Eligible for Prior Learning, Assessment and Recognition:	Yes
Originated by:	Bhim Harlal, Sujeet Lohan
Revised by:	Ilia Nika
Revision Date:	Fall 2014
Current Semester:	Fall 2015
Approved by:	p pesikan c/o Chairperson/Dean

Students are expected to review and understand all areas of the course outline.

Retain this course outline for future transfer credit applications. A fee may be charged for additional copies.

This course outline is available in alternative formats upon request.

Course Description

This is a survey course that introduces students to the common terminologies, methodologies, tools, and techniques used in software engineering.

It gives students a strong foundation which forms the basis for advanced studies of software engineering. Fundamental concepts of related topics such as: team and teamwork, risk management, software testing, and software engineering code of ethics and professional practices are also introduced.

Program Outcomes

Successful completion of this and other courses in the program culminates in the achievement of the Vocational Learning Outcomes (program outcomes) set by the Ministry of Training, Colleges and Universities in the Program Standard. The VLOs express the learning a student must reliably demonstrate before graduation. To ensure a meaningful learning experience and to better understand how this course and program prepare graduates for success, students are encouraged to review the Program Standard by visiting http://www.tcu.gov.on.ca/pepg/audiences/colleges/progstan/. For apprenticeship-based programs, visit http://www.collegeoftrades.ca/training-standards.

Course Learning Outcomes

The student will reliably demonstrate the ability to:

1. Describe and clarify terms widely used in software engineering.
2. Apply software engineering techniques and methodologies to solve simple problems.
3. Assess the importance of teamwork in software engineering.
4. Describe the essentials of team building, and building of team spirit.
5. Determine the types of risks associated with software development and suggest ways to manage these risks in order to mitigate their negative impacts
6. Explain the various types of software tests and explain how and why each is used.
7. Assess the importance of software engineering code of ethics and professional practice in making software engineering a beneficial and respected profession (ACM/IEEE-CS, and Canadian Information Processing Society (CIPS) codes of ethics).
8. Relate software engineering Code of Ethics and Professional Conduct to Global Citizenship and Equity initiatives of equity, social justice, respect, and care for human well-being at the local, national, and global levels.
9. Apply a variety of tools (word processor, spreadsheet, presentation tool e.g. Power Point, charting and diagramming tool e.g. Visio) to produce diagrams, and descriptive, mathematical, graphical models that represent simple problems and solutions

Essential Employability Skills (EES)

The student will reliably demonstrate the ability to*:

1. Communicate clearly, concisely and correctly in the written, spoken, and visual form that fulfills the purpose and meets the needs of the audience.
2. Respond to written, spoken, or visual messages in a manner that ensures effective communication.
3. Execute mathematical operations accurately.
4. Apply a systematic approach to solve problems.
5. Use a variety of thinking skills to anticipate and solve problems.

6. Locate, select, organize, and document information using appropriate technology and information systems.
7. Analyze, evaluate, and apply relevant information from a variety of sources.
8. Show respect for diverse opinions, values belief systems, and contributions of others.
9. Interact with others in groups or teams in ways that contribute to effective working relationships and the achievement of goals.
10. Manage the use of time and other resources to complete projects.
11. Take responsibility for one's own actions, decisions, and consequences.

Note: *There are 11 Essential Employability Skills outcomes as per the Ministry Program Standard. Of these 11 outcomes, the following will be assessed in this course.*

Global Citizenship and Equity (GC&E) Outcomes

N/A

Text and other Instructional/Learning Materials

Text Book(s):

BNDL:Hunt, M. & Waxer, B. 2011. Microsoft Office 2013 illustrated (Fundamentals) and Video DVD for MS Office 2013: Introductory. Course Technology.

ISBN : 1285269276

Online Resource(s):

http://proquestcombo.safaribooksonline.com/book/office-and-productivity-applications/9780735678293

Material(s) required for completing this course:

Study Notes, Hand outs, Power Point slides,Web site references etc.

Evaluation Scheme

- Lab Assignments: Lab assignments based on the course contents.
- Group Assignment: It is an assignment to be completed by a group of 2 students and based on course contents.
- Term Tests: Term Test 01. It has two components - Written part (10%) and Lab part (10%)
  Term Test 02. It has two components - Written part (10%) and Lab part (10%)
  Term Test 03. It has two components - Written part (10%) and Lab part (10%)
- Project Work: Project work. It has 10% weightage.

Evaluation Name	CLO(s)	EES Outcome(s)	GCE Outcome(s)	Weight/100
Lab Assignments	1, 2, 3, 4, 5, 6, 7, 8, 9	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11		15
Group Assignment	3, 4	1, 2, 6, 8, 9, 10		10
Term Tests	1, 2, 3, 4, 5, 6, 7, 8, 9	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11		60
Project Work	1, 2, 3, 4, 5, 6, 7, 8, 9	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11		15
Total				100%

If students are unable to write a test they should immediately contact their professor or program Chair for advice. In exceptional and well documented circumstances (e.g. unforeseen family problems, serious illness, or death of a close family member), students may be able to write a make-up test.

All submitted work may be reviewed for authenticity and originality utilizing Turnitin®. Students who do not wish to have their work submitted to Turnitin® must, by the end of the second week of class, communicate this in writing to the instructor and make mutually agreeable alternate arrangements.

When writing tests, students must be able to produce official College photo identification or they may be refused the right to take the test or test results will be void.

Student Accommodation

It is College Policy to provide accommodation based on grounds defined in the Ontario Human Rights Code. Accommodation may include modifications to standard practices. Students with disabilities who require academic accommodations must register with the Centre for Students with Disabilities. Students requiring accommodation based on other human rights grounds should talk with their professors as early as possible. Please see the Student Accommodation Policy.

Use of Dictionaries

Program or School Policies

N/A

Course Policies

n/a

College Policies

Students should familiarize themselves with all College Policies that cover academic matters and student conduct.

All students and employees have the right to study and work in an environment that is free from discrimination and harassment and promotes respect and equity. Centennial policies ensure all incidents of harassment, discrimination, bullying and violence will be addressed and responded to accordingly.

Academic honesty is integral to the learning process and a necessary ingredient of academic integrity. Academic dishonesty includes cheating, plagiarism, and impersonation. All of these occur when the work of others is presented by a student as their own and/or without citing sources of information. Breaches of academic honesty may result in a failing grade on the assignment/course, suspension or expulsion from the college.

For more information on these and other policies, please visit www.centennialcollege.ca/about-centennial/college-overview/college-policies.

Students enrolled in a joint or collaborative program are subject to the partner institution's academic policies.

PLAR Process

This course is eligible for Prior Learning Assessment and Recognition (PLAR). PLAR is a process by which course credit may be granted for past learning acquired through work or other life experiences. The PLAR process involves completing an assessment (portfolio, test, assignment, etc.) that reliably demonstrates achievement of the course learning outcomes. Contact the academic school to obtain information on the PLAR process and the required assessment.

Topical Outline (subject to change):

Week	Topics	Readings/Materials	Weekly Learning Outcome(s)	Instructional Strategies	Evaluation Name	Evaluation Date
1-2	Computer Fundamentals 1.Four basic operations of computers (Information Processing Cycle): Input, Process, Output, Storage 2.Bits and Bytes, Units of binary storage 3.Data vs information 4.Components of Information systems: 5.Hardware, Software, Data , People and Procedures Definition of Software: 1.Systems Software: Operating Systems, Device Drivers, Utilities, Program Languages, Compilers, Interpreters 2.Application Software: Office Productivity Software, Examples of other applications software Definition of Hardware 1.Central Processing unit (CPU), Arithmetic and Logic Unit, Control Unit , Registers, Main/Primary Memory (RAM), Secondary (Auxiliary) Memory. 2.Basic computer architecture	Lectures, Handouts and Web Sites	•Explain the information processing cycle. •Explain the components of an information system •Differentiate between Data and information •Distinguish between Application Software and Systems Software •Distinguish between main (primary) memory and secondary (auxiliary) storage. •Describe the essential components of the CPU and their functions •Explain a basic computer architecture	Lecture Demonstration Lab Session Group Discussion/Debate	Complete-Concepts Review and one of the Independent Challenge exercises	

Week	Topics	Readings/Materials	Weekly Learning Outcome(s)	Instructional Strategies	Evaluation Name	Evaluation Date
3	Software Engineering Terms and Practices: 1.Computer Science Vs Software Engineering 2.Engineering Practices involved in developing software products: Software Engineering 1.Definition 2.Functions of Software Engineering Project Management 1.Definition 2.Functions of Project Management Reasons for failures in Software Engineering Projects. Five important Software principles and practices 1.Abstraction 2.Models and Modelling (Purpose of models, Data Models, Process Models, Object Models) 3.Patterns 4.Reuse 5.Methodologies Relationships of models tools and techniques in a system development methodology	(Standish Group Report) Lectures, Handouts and Web Sites	• Differentiate between computer science and software engineering • Explain the engineering practices in developing software products. • Define and list the functions of software system engineering, software engineering, and Project management. • Explain the term software development methodology and list the engineering practices involved in developing software projects • Explain the Differences among Model, Tool, Technique and Methodology • Explain the purpose of Models • Differentiate among Data Model, Process Model and Object Model • Explain and explain the importance of abstraction models, patterns, reuse and methodologies in Software development	Lecture Demonstration Lab Session Group Discussion/Debate	Windows7/8-Unit A and B Complete Concepts review and Skills Review. Complete one of the independent Challenge exercises. Office Word 2013: Unit C and D Complete Concepts review and Skills Review. Complete one of the independent Challenge exercises	
4	1.Modelling Language 2.Syntax 3.Semantics 4.Brief descriptions	Lectures, Handouts and Web Sites	•Explain the term: syntax, and semantics as applied to a modeling language •Explain models used for System components and for managing project development	Lecture Demonstration Lab Session Group	Lab Assignment 01 – Out (5%)	

Week	Topics	Readings/Materials	Weekly Learning Outcome(s)	Instructional Strategies	Evaluation Name	Evaluation Date
	and examples of models used for: 5.System Components using UML 6.Managing Project Development 7.Software Development Tools 8.Project management tools e.g. Microsoft Project 9.Charting and Diagramming tools e.g. Microsoft Visio 10.Tools to develop and maintain complex information systems: Computer-Aided Software Engineering (CASE) Tools e.g. Rational Rose, Visible Analyst, Visual Paradigm etc. 11.Software recovery and rejuvenation processes: Forward engineering, reverse engineering, round-trip software engineering, re-engineering. 12.Project plan, project scope, project deliverables and project milestones, work breakdown structure, project schedule 13.Brief description and examples of models used to support project scheduling: Gantt		•List software tools commonly used to support Project Management, Modeling and for developing and maintaining complex information systems. •Explain the purpose and application of project Management Tools, Charting and Diagramming Tools and CASE Tools. •Explain the term Technique and explain its usage in Software Engineering •Explain the relationships among models, tools, and techniques •Explain the importance of forward engineering, reverse engineering, round-trip engineering, re-engineering in Software recovery and rejuvenation. •Explain what a good project plan contains •Explain the term work-breakdown structure and its importance •Explain the importance of project deliverables, project milestones and project schedule in a software project development	Discussion/Debate. Office Word 2013: Unit E and F Complete Concepts review and Skills Review. Complete one of the independent Challenge exercises		

Week	Topics	Readings/Materials	Weekly Learning Outcome(s)	Instructional Strategies	Evaluation Name	Evaluation Date
	chart and PERT chart.					
5	Software Development Life-Cycle Models: 1.Predictive approach and adaptive approach to software development and associated risks levels 2.Waterfall Model: a.(Structured analysis techniques to analyse and develop data and process models of information systems) b.Phases and deliverables of the waterfall model c.Drawbacks of the waterfall model	Lectures, Handouts and Web Sites	•Differentiate between predictive approach and adaptive approach to software development •Explain the principles of the Waterfall model. •Explain the phases and deliverables of the Waterfall model •Explain the drawbacks of the Waterfall Model	Lecture Demonstration Lab Session Group Discussion/Debate	Lab Assignment 01 – Due TERM TEST-01(Written - 10%, Lab-10%)	
6	Software Development Life-Cycle Models continued: 1.Unified Process (UP) a.(Object-oriented techniques used to analyze, document and model information systems) b.Phases, disciplines, and iterations c.Iterations as a means of risk control	Lectures, Handouts and Web Sites	•Explain the phases of the Unified Process (UP) and the various UP disciplines •Explain why the UP disciplines are used in varying amounts in each iteration • Explain how iterations are used as means of controlling risks	Lecture Demonstration Lab Session Group Discussion/Debate	Excel 2013: Unit G Complete Concepts review and Skills Review. Complete one of the independent Challenge exercises Group Assignment-Out(10 %)	
7	Current trends in software systems development 1.Agile development philosophy, manifesto, modelling and principles	Lectures, Handouts and Web Sites	•Explain the Agile development philosophy •Explain Agile modeling principles and practices •Compare and contrast Extreme Programming and Scrum development. •Explain the Crystal family of methodologies (Crystal Clear and Crystal Orange)	Lecture Demonstration Lab Session Group Discussion/Debate	Lab Assignment 02 – Out (5%) Excel 2013: Unit H and I Complete	

Week	Topics	Readings/Materials	Weekly Learning Outcome(s)	Instructional Strategies	Evaluation Name	Evaluation Date
	2.Extreme Programming (XP) 3.SCRUM 4.Crystal family of methodologies (Crystal Clear and Crystal Orange) 5.Microsoft Solutions Framework (MSF) for agile development		•Explain the functional principles of Microsoft Solutions Framework (MSF)		Concepts review and Skills Review. Complete one of the independent Challenge exercises	
8	Characteristics associated with good-quality software (brief description only): 1.Correctness, Reliability, Performance, Integrity, Usability, Maintainability, Testability, Portability, Re-usability, Robustness, Security Software component Standards 1.Common Object Model Plus (COM+) 2.Enterprise Java Beans (EJB) 3.Simple Object Access Protocol (SOAP) and .NET	Lectures, Handouts and Web Sites	•Explain the various characteristics that are associated with good-quality software. •Recognize that some of these qualities can be mutually contradictory (e.g. if High Performance is required, Portability will probably suffer) •Recognize that not every attribute is desirable in every piece of software •Explain the importance of software components •Describe briefly component interaction standards such as COM+, EJB, SOAP. and .NET	Lecture Demonstration Lab Session Group Discussion/Debate	Lab Assignment 02 – Due Power Point 2013: Unit M and N Complete Concepts review and Skills Review. Complete one of the independent Challenge exercises	
9	1.Software Requirements Specifications 2.Functional Requirements 3.Non-functional Requirements IEEE Standards of Software Requirements Specifications (IEEE	Lectures, Handouts and Web Sites	•Identify the functional and non-functional requirements of a system •Recognize that functional requirements specify functions a system or system component must be able to perform and that can be tested. •Differentiate between functional and non-functional requirements •Explain and apply the IEEE standard for software requirements specification	Lecture Demonstration Lab Session Group Discussion/Debate Integration 2013: Unit O Complete Concepts review and Skills Review. Complete one of the independent	Group Assignment - DUE Lab Assignment 03 – Out (5%) Group Project - (15%) Guidelines/Topics	

Week	Topics	Readings/Materials	Weekly Learning Outcome(s)	Instructional Strategies	Evaluation Name	Evaluation Date
	830-1933)			Challenge exercises Understanding Office in the Cloud – Office 2013 Appendix		
10	Software Development Team 1.Challenges of Software Development and Roles of Teams 2.Management issues arising from team-based software development 3.Prerequisites of team building Ways of organizing teams: -Cooperating peers -Chief developer -Collaborative specialist 4.Team structure and task assignment	Lectures, Handouts and Web Sites	•Explain the challenges of large software projects and the need for teamwork •Explain the management issues arising from team-based software development •Describe the different ways of organizing teams •Explain the common principles of team organization	Lecture Demonstration Lab Session Group Discussion/Debate MS Visio 2013 Environment: Template, stencils, shapes. Using tools from Standard toolbar, and Drawing toolbar to create and edit shapes Adding, sizing, rotating, and copying shapes. Searching for shapes. Connecting shapes	TERM TEST-02 (Written -10%, Lab-10%) Lab Assignment 03 – Due Group Project members finalized.	
11	Software Testing 1.Unit (Component) Testing 2.Integration Testing 3.Systems Testing 4.Usability Testing 5.Performance Test: Response Time, Throughput 6.User Acceptance Testing 7.Alpha Testing 8.Beta Testing Verification and Validation (IEEE definition)	Lectures, Handouts and Web Sites	•Explain the various types of software tests and explain how and why each is used. •Differentiate between Alpha testing and Beta testing •Define the terms: Verification and Validation •Differentiate between validation and verification and explain the importance of these two activities throughout the software engineering process	Lecture Demonstration Lab Session Group Research and Presentation	Visio cont’d: Formatting and adding color to shapes and diagrams. Creating Text block. Selecting multiple objects. Aligning and Distributing shapes. Creating organizational charts, and	

Week	Topics	Readings/Materials	Weekly Learning Outcome(s)	Instructional Strategies	Evaluation Name	Evaluation Date
					flow charts. Visio exercises to create diagrams relating to Software engineering	
12	Professional Practice and Ethical Conduct Software Engineering Code of Ethics and Professional Practice (IEEE_CS/ACM , and CIPS) Project Risk and Risk Management 1. Definition of Risk 2. Importance of Project Risk Management 3. Activities of risk Management: a. Risk Assessment b. Risk Identification c. Risk Analysis d. Risk Prioritization 4. Risk Control a. Risk reduction b. Risk Management Planning c. Risk resolution and control 5. Feasibility and Risks: Risks associated with Technological feasibility, Schedule feasibility, Resource feasibility, Economic feasibility.	Lectures, Handouts and Web Sites	•Explain the moral and ethical principles that guide software engineers in their practices. •Explain the importance of software engineering code of ethics and professional practice in making software engineering a beneficial and respected profession. •Explain the term risk •Explain the objectives of risk management •Describe the activities associated with risk management •Explain the risk associated with performing technological, schedule, resource, and economic feasibility	Lecture Demonstration Lab Session Group Research and Presentation	Work on Group Project Report and Presentation	

Week	Topics	Readings/Materials	Weekly Learning Outcome(s)	Instructional Strategies	Evaluation Name	Evaluation Date
13-14	Term Project Review and Presentation	n/a	n/a	n/a	Week-13 -> Group Project Presentation, Report, Participation – 15% Week-14 -> Term Test 03(Written – 10%, Lab - 10%)	