COURSE SYLLABUS

Course Number:	CS123		
Title:	Introduction to Software Engineering		
Department/Program:	DISCS	**School:**	Science and Engineering
Semester:	First Semester	**School Year:**	2015-2016
Instructor/s:	Marlene M. De Leon, Ph.D. Guillermo Paolo M. Agloro, M.S.		

A. COURSE DESCRIPTION

This course provides an overview of the software engineering process. Topics include requirement analysis, analysis modeling, software design fundamentals, software testing, quality assurance, software maintenance, and object-oriented software engineering.

B. COURSE OUTCOMES

By the end of this course, students should be able to:

- Compare various System Development Life Cycle approach
- Plan and estimate time and cost of a software development project
- Construct UML diagrams
- Apply an object oriented methodology and agile process to analyze and solve a real-world problem
- Write a well-designed, coherent, requirements and design specification
- Develop actual software with accompanying documents

C. COURSE OUTLINE / TIMEFRAME

WEEK/ TOPIC	LEARNING OBJECTIVES	ACTIVITIES	STUDENT OUTPUT
Weeks 1-2: Introduction to Software Engineering	Identify characteristics of software. Describe software quality. Explain why software engineering is important. Discuss the most probable reasons for software development success and failure. To distinguish between classical and modern approaches in software engineering.	Lecture & Discussion	Selection of Team members Short Project Proposal Recitation
Weeks 2-3: Teams Management	Explain various types of team organizations. Select appropriate team organization for the project.	Lecture & Discussion Quiz	Selection of Team Organization & Member roles Recitation Answers to quiz
Weeks 4-5: Software Planning & Project Management	Explain the main tasks undertaken by project managers Describe software project management and to identify its distinctive characteristics Discuss project planning and the planning	Lecture & Discussion Lab Practice Quiz	Estimate of budget & schedule Risk list Answers to quiz

	process Explain how graphical schedule representations are used by project management Plan the software process Estimate duration and cost Discuss the notion of risks and the risk management process		
Week 5-6: Software Process & Life Cycle	Discuss the characteristics, strengths & weaknesses Software Life Cycle (SLC): • Code-and-fix Waterfall • Rapid prototyping • Iterative & Incremental • Agile and XP • Open-source • Synchronize-and-stabilize • Spiral	Lecture & Discussion Lab Practice: User stories Lab Practice: Agile Estimation	Choice SLC to be used User Stories Working Code of the first iteration (GUI prototype)
Week 7-8: Requirement Engineering	Discuss important principles in requirement engineering Identify requirement process steps Develop use case diagram	Lecture & Discussion	Recitation Working code of the second iteration (input validation, automatic testing) Use cases & use case diagram
Weeks 8-9: UML Diagrams	Discuss principles and reasons for using UML diagrams To develop UML diagram • Class diagram • Interaction diagram • State chart diagram • Activity diagrams • Packages diagram • Component diagram • Deployment diagram Review UML diagrams Relate UML and iteration & incremental SLC	Lecture & Discussion Quiz	Recitation Set of UML diagrams (minimum of class diagram, and interaction or activity diagram) Answers to quiz
Weeks 10-11: Software Analysis	Describe the importance need of specifications document Apply structured systems analysis using Data Flow Diagram (DFD) and Entity-relationship modeling (ERM) Describe formal specifications using Finite state machines and Petri nets Use Noun Extraction, CRC card and boundary extraction of classes for object-oriented analysis	Lecture & Discussion Midterm Exam	Recitation Specification Document. Answers to the midterm exam
Weeks 11-12: Software Design	Distinguish module from object Compare level of Cohesion and Coupling Discuss data encapsulation,	Lecture & Discussion	Recitation Design Documents

	Abstract data types, Information hiding, Object Inheritance, polymorphism, abstraction, and dynamic binding Distinguish Operation-oriented design, Data-oriented design and Object-oriented design		
Weeks 13-14: Software Implementation	Justify reasons for choice of programming language Apply good programming practice for code reuse and coding standard	Lecture & Discussion Quiz	Recitation Working code of the third iteration (object-oriented code, classes, interface) Answers to quiz
Weeks 14-15: Software Testing, Maintenance & Quality Assurance	Distinguish between Non-execution-based testing and Execution-based testing Describe Capability Maturity Models (CMM) Compare various testing methods • Black-box unit-testing • Glass-box unit-testing • Code walkthroughs • Clean room • Integration testing • Product testing • Acceptance testing	Lecture & Discussion Lab Practice	Recitation Working code of the fourth iteration (abstract classes, refactoring, error free code, developer guide documentation)
Weeks 15-17: Reusability & Portability	Discuss the concept of reuse during design and implementation and maintenance Describe the need and techniques for achieving portability Discuss the aspects and requirements of post-delivery maintenance including Software Versions, Configuration Control, Management of maintenance, and Reverse engineering	Lecture & Discussion Quiz	Recitation Working code of the fifth iteration (executable, auto deployment, error free code, user guide documentation)
Week 18: Final Exam	Describe important concepts covered during the second half of the semester	Final Exam	Final Exam

D. REQUIRED READING

Schach. *Object Oriented and Classical Software Engineering (8th ed).* McGraw Hill, 2010.
Saleh. *Software Engineering*, Cengage Learning Asia, 2010.

E. SUGGESTED READINGS

Vliet, *Software Engineering: Principles and Practice 3rd ed*, 2008
Sommerville, *Software Engineering 10th Edition,* 2015
Booch, Rumbauch & Jacobson, *The Unified Modeling Language user guide*, Addison Wesley, 1999
Laplante, *What every engineer should know about software engineering*, CRC Press, 2007
Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*, Pearson Education, Inc., 2009

F. COURSE REQUIREMENTS

Project	40%
Final Exam	20%
Midterm Exam	20%
Quizzes / HW / Exercises	10%
Recitation & Participation	10%

Project

- The main objective of this exercise is to allow the students to simulate a real world project from the definition of requirement, analysis to design specification stage, implementation with appropriate documentation using Use Case, Class diagram and other UML diagrams.
- Select a creative project that will be assigned in the class.
- Select any combination of life cycle model (with reasoning) with at least four iterations of implementation (see student's output in course outline and timeframe above)
- Students are free to use any programming language for the implementation.
- Students are allowed to group themselves in teams of three (3) to five (5). Groupings to be submitted are considered permanent.
- Evidence of collaboration must also be submitted together with PDF source of literatures and Certificate of Authorships.
- Submission due:
  - Deliverable of milestone (throughout the course)
  - 5 minutes presentation of milestone (throughout the course)
  - Final presentation one week before the Final Exam
  - Final Report (must be submitted at the Final exam)
- Final report should contain
  - life cycle model to be used & reasoning
  - requirement: Use Case
  - analysis: UML
  - design: detail specification (UML)
  - Testing (plan, actual and results)
  - implementation: working code, testing code for each iteration
  - final documentation
  - break down cost & time estimation vs actual
  - Role and contribution of each team member
  - lessons learned

G. GRADING SYSTEM

93-100	A	Excellent
87-92	B+	Very Good
81-86	B	Good
75-80	C+	Satisfactory
69-74	C	Sufficient
60-68	D	Passing
<60	F	Failure

Notes:

- Rounding off of grades is at the discretion of the instructor. Rounding off of grades is not automatic (even if the grade is x.9999999.).
- No exemptions will be given for the final exam.

H. CLASSROOM POLICIES

1. Attendance will be checked by the class beadle every meeting. If you exceed your allowable cuts for the semester, you may be given a W grade.

2. In case you cut, it will be your responsibility to know the material covered for the day. The instructor also reserves the right to give unannounced quizzes or graded lab exercises at any time.

3. No make-up tests will be given unless you have a very valid reason, and proof supporting it

4. Playing games is strictly prohibited during class hours. Web browsing and reading email are also prohibited, unless done in connection with the current lecture or lab topic *and* allowed by the teacher.

5. Cheating will not be tolerated. Cheating in any requirement will result in a *minimum* penalty of having a grade of 0 for that requirement, and will be reported to the appropriate authorities, as provided for by the Student Handbook. Duplicate work will merit penalties for *both* the student who copied and the student from whom the work was copied.

6. Students are expected to comply with the DISCS Academic Integrity Policy. With each submission, students must include a certification that their work is substantially their own and not copied from others. In addition, students must clearly acknowledge and specify any help from outside sources such as other classmates, the Web, books, etc., that they received while doing their projects. Failure to acknowledge such may be interpreted as intellectual dishonesty. Consult the course website for details on these policies.

7. Additional policies, with due consultation with the students, may be implemented by the instructor to adapt to the class environment.

I. CONSULTATION HOURS

Marlene M. De Leon, Ph.D. TTh 0800-1000 @ F208 I would appreciate an appointment is set in advance. You may contact me via the following: Email: mmana@ateneo.edu Phone: 426-6001 loc 5660	**Guillermo Paolo M. Agloro, M.S.** MWF 1600 – 1730 @F208 I would appreciate an appointment is set in advance. You may contact me via the following: Email: pagloro@ateneo.edu Phone: 426-6001 loc 5660