CSC 260 – Software Engineering
Summer 2013 (May 20 – July 12)

Professor:	Dr. S. Monisha Pulimood
Meeting Times:	Thursdays 9:00 a.m. – 12:30 p.m. in class. (Starts May 23.)
Meeting Room:	Holman Hall 370
Office:	Holman Hall 225
Office Hours:	Tuesdays 6:00 – 7:00 p.m., online via Skype or Google Hangout; Thursdays 12:30 – 2:00 p.m., in person in Holman Hall 225; By appointment, in person, via Skype or Google Hangout.
E-mail address:	pulimood [at] tcnj [dot] edu
Google id:	monishatcnj [at] gmail [dot] com
Skype id:	monishapulimood
Github id:	tcnjcs / pulimood [at] tcnj [dot] edu
Course Website:	http://tcnjcsc340.pbworks.com (Access is limited to students enrolled in Summer 2013.)
Piazza:	https://piazza.com/tcnj/summer2013/csc260/home (Access is limited to students enrolled in Summer 2013.)
Lucidchart:	https://www.lucidchart.com/e/pulimood.tcnj.edu (Access is limited to students enrolled in Summer 2013.)

Office Hours and Contact:

I will be available in my office, Holman Hall 225, for meetings in person, without appointment during my posted office hours, Thursdays 12:30 – 2:00 p.m.

I will be available online on Tuesdays 6:00 – 7:00 p.m. via Skype (www.skype.com) or Google Hangout (www.google.com/+/learnmore/hangouts/).

I can also be available at other times by appointment for meetings in person in my office, or via Skype or Google Hangout. Please email me to make an appointment to meet at times other than office hours.

You may post questions on Piazza that either your classmates or I can respond to. I will typically respond to emails and posts on Piazza within a few hours, but no more than 24 hours.

Course Units:

1 unit (4 credits).

CSC 260 – Software Engineering
Summer 2013 (May 20 – July 12)

Course Prerequisites:
CSC 230 or CSC 250 with a minimum grade of C.
The prerequisite courses provide students with a basic understanding of object-oriented concepts and programming languages that are crucial to the mastery of the material to be covered in this course.

Course Attributes:
This course counts towards the advanced core requirement for Computer Science and meets the mid-level writing requirement for Computer Science.

Course Description:
This course introduces students to software engineering with an emphasis on collaborative problem solving and software application development. It aims to integrate the principles of computer science and other disciplines to develop systematic models and reliable techniques for producing high-quality software. Advanced object-oriented and implementation concepts are introduced through the C++ programming language. The Unified Modeling Language is used to describe each stage of application development, particularly analysis and design.

This course is part of a NSF-sponsored project (Award # 1141170) to determine how to better engage undergraduates in computer science. To this end, students enrolled in this course will participate in assessment of projects, learning goals, and the like. Participation in these assessment procedures is not required and does not affect the grade you earn in this class. However, it will help shape curriculum development nationwide, and is thus, both valued and important. Additional information on this project will be provided to students on the first day of class.

Blended Learning Format:
This course uses a blended learning (BL) format that combines face-to-face teaching and learning sessions with asynchronous learning activities. The class will meet face-to-face once a week; in addition, you will complete assigned activities during each week of the course. While much of your work will be carried out asynchronously, some activities may be conducted online, such as conference sessions, collaborative efforts, and discussion. As such, you will need to have daily access to the Internet.

The advantage of the BL format is that you can make progress on your assignments and readings in a location and at times that are most convenient for you. Although a BL course may seem easier than a regular course due to the reduced number of class meetings, less time spent in on-campus classes does not mean less work during the course; you cannot adopt the mentality that you only need to work one day per week. Since you must do the same amount of work as in a regular course but without the structure that traditional interactions provide, the BL format places an extra burden on you to be especially disciplined. You must assume more responsibility for managing your learning time, and distribute your readings and assignments over the span of the week so that you do not fall behind. Your active participation is essential for a rich learning experience.

CSC 260 – Software Engineering
Summer 2013 (May 20 – July 12)

Learning Goals:

The primary goals of this course are for students to understand, internalize and become proficient in the following core computer science concepts:

Computer Science Foundations: use and support for abstraction; concept of a system; human factors; code reuse and libraries; implementation and run-time issues; defensive programming.

Software Modeling and Analysis: modeling principles such as decomposition, generalization, etc.; pre- and post-conditions; behavioral modeling using state diagrams, use case analysis, interaction diagrams, etc.; requirements process; eliciting requirements (techniques and sources); documentation.

Software Design: fundamental design issues and trade-offs; design principles such as information hiding, cohesion, and coupling; object-oriented design strategy; human computer interface design.

Construction Tools: development environments, versioning systems, presentation tools, and project management tools.

Group Dynamics and Communication Skills: dynamics of working in teams / groups; dealing with uncertainty or ambiguity; reading, understanding, and summarizing reading (source code, documentation); writing (assignments, reports, evaluations, justifications); team and group communication (both oral and written); presentation skills.

Assignments and a collaborative project serve to reinforce principles and concepts learned in class while providing hands-on experience with related software tools. Problem solving skills will be further developed. Written and oral reports broaden the students' knowledge base, while enhancing presentation, communication, and technical writing abilities.

Students will be able to apply the understanding and skills developed to courses that follow this one in the curriculum sequence. Many of these are vital skills for today's demanding workplace.

Course Materials – Textbooks:

Required Textbooks: A student must have personal copies of both "Required" textbooks listed below, since exercises, homework, readings and assignments will frequently be taken from them.

Students may use the print or electronic versions of textbooks. Please be sure to check terms of service prior to purchase since electronic books are often available for limited periods and cannot be 'sold' back.

1. "*Programming and Problem Solving with C++*", 6e, Comprehensive Edition
   Authors: Nell Dale and Chip Weems
   Publisher: Jones & Bartlett; ISBN-13: 9781284028768
   http://www.coursesmart.com/IR/1430375/9781284032796?__hdv=6.8.

2. "*Software Engineering: A Practitioner's Approach*", 7th Edition
   Author: Roger S Pressman, R.S. Pressman & Associates
   Publisher: McGraw-Hill; ISBN-13 9780073375977, MHID 0073375977
   http://www.coursesmart.com/IR/1430375/0077279379?__hdv=6.8.

Recommended Books: You are strongly advised to have a personal copy of "Recommended" books listed below, but these are not mandatory.

"*A Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet*", Third Edition by Sara Baase;
ISBN-10 0-13-600848-8, ISBN-13 978-0-13-600848-4
Publisher: Prentice Hall
http://www.coursesmart.com/IR/1430375/9780136008859?__hdv=6.8

"*Practical Tools and Techniques for Software Development (2nd Edition): A Computer Science/Information Technology curriculum companion*" by Edward L Crookshanks
ISBN-10: 147514153X, ISBN-13: 978-1475141535
Publisher: CreateSpace Independent Publishing Platform; 2 edition (April 3, 2012)
PDF available from lulu.com; Print available from Amazon.com.

Reference Books: These books are useful as additional reference for specific topics.

"*C++ Coding Standards*" by Herb Sutter and Andrei Alexandrescu.

"*Object-Oriented Software Engineering*", by Stephen Schach

"*Absolute C++*" (3rd edition) by Walter Savitch

Course Materials – Management and Software:

- Course information, including this syllabus, assignments and slides, will be available on Canvas.
- Information will be provided as announcements in class, or posted on Canvas or Piazza, or sent via email to enrolled students' TCNJ email addresses. It is the student's responsibility to keep up to date by frequently checking these.
- Students will submit assignments in the appropriate Canvas Dropbox. Grades and feedback on assignments will be posted on Canvas.
- Final grades will be posted on PAWS.
- Class collaboration will be through Canvas, the course wiki and Piazza.
- A C++ aware IDE such as Eclipse or XCode should be used for writing source code.
- Programs should be compiled using g++.
- Github should be used for project management and source code control. You can create a free account with private repositories using your TCNJ email id.
- UML diagrams should be produced using Lucidchart.
- All required software will be made available on the machines in Holman Hall 370 or free of charge on the Internet.
- Selected software, such as Visio and Visual Studio, will be available (free of charge) for download to students' personal machines.

Students are expected to independently learn to use the appropriate software tools.

CSC 260 – Software Engineering
Summer 2013 (May 20 – July 12)

Course Requirements and Student Assessment:

The field of computer science is continually evolving, so it is essential that students are well prepared to adapt quickly and efficiently to change and also to perform their jobs in a productive manner. Through the activities in this course, students will begin to build a strong foundation in software engineering. A more important outcome is that students will learn how to work collaboratively with others and to integrate concepts from computer science and other disciplines in order to solve problems and implement solutions.

Depending on the material to be covered, different activities and modes of teaching will be used, including group exercises, discussions, traditional lectures, lab work and tests. To pass the course with a C or better, a student must demonstrate strong understanding of the course material. The course grade will be computed as the weighted average as follows:

	Category	**Weight**
1.	Participation and Preparation	15%
2.	Writing and Programming Assignments	30%
3.	Collaborative Project	25%
4.	Quizzes	15%
5.	Final Evaluation	15%
	Total	**100%**

1. **Participation and Preparation:**
   Students are expected to abide by the College policy on attendance (available at http://policies.tcnj.edu/policies/digest.php?docId=8162). Students are expected to attend and be punctual for all classes. When possible, a student must inform me ahead of time about any planned absences, e.g. for a field trip in another class or to attend a conference. Students are responsible for making arrangements to catch up on missed work. There is no direct point-penalty for absenteeism, but chronic absenteeism will affect the grade indirectly since the student may miss in-class collaborative exercises and will lose the benefit of class discussions and presentations.

   Students are expected to participate actively in the learning experiences to gain the maximum benefits. This includes coming prepared to class, having reviewed material covered in the previous class and to be covered in the current class; actively engaging in class discussions; completing class and homework exercises in a timely manner; etc.

   Students will be required to complete a blended learning student feedback survey via Qualtrics in addition to the TCNJ Student Feedback on Teaching form.

   Students will work on in-class and homework exercises, individually or in small groups, to better understand concepts. Exercises that are not completed in class are to be completed before the next class.

   Students must assume responsibility for managing their own time, and distribute readings, assignments, and project work over the span of the week so that they do not fall behind. The Course Schedule and Assignment Schedule are available on Canvas and will be updated as needed to aid in time management.

2. Assignments:

There will be two to three large programming assignments and two to three writing assignments. Details will be posted at the appropriate times on the wiki. Assignments will be due as per the schedule.

The requirements of the mid-level writing intensive course (twenty double-spaced pages of finished prose) will be achieved by students individually completing:

- Programming assignments that require several (more than ten) pages of source code each, as well as a minimum of two pages of maintenance documentation that includes well-written comments describing the purpose of the assignment, the purpose of various code fragments, and pre- and post-conditions for each function.
- Detailed graphical/textual descriptions of the analysis and design of the system to be developed for the programming assignments.
- Technical reports (total of about fifteen double-spaced pages) on topics related to the course material. The writing style, citation of references, and bibliography for reports must conform to guidelines for articles submitted to scientific journals, like those published by ACM.

All assignments in this category must be completed and submitted. The student will automatically be assessed an 'F' grade for the course if there are any incomplete or missing assignments.

Grading and Feedback Procedure:

- An assignment must be complete for it to be eligible for a passing grade. Incomplete assignments will be returned without being graded.
- Programs that do not compile will automatically get a grade of 0.
- Rubrics will be provided for assignments.
- Grades and feedback will be posted on Canvas.
- Students are strongly encouraged to utilize feedback to improve future submissions.
- There is a 10% penalty for each day an assignment is late.
- After the grade for an assignment has been posted, the student may resubmit it once for re-grading if it has been significantly improved. This option must be exercised within 2 weeks of receiving the graded assignment.
    - The revised grade will be the average of the two submissions.
    - Late penalties on the original submission will apply to the resubmission.

3. Collaborative Projects:

Students will collaborate in teams to complete a semester-long project. Details for the project will be posted on Canvas.

- The collaborative project is completed in several phases. The materials for each phase may be improved or modified as per feedback provided or to reflect changes as the project evolves. These are then to be synthesized and resubmitted for grading as part of the final project report.
- Each team will make an oral presentation of the project to the entire class. The presentation will include most components of the project report and a demonstration of the working final product.

4. **Quizzes:**
   Throughout the semester there will be announced or unannounced quizzes on material covered in the preceding week. Quizzes will be administered online through Canvas.

5. **Final Evaluation:**
   As per the College final examination policy (available at http://policies.tcnj.edu/policies/digest.php?docId=6242), the final evaluation will be comprehensive and integrate concepts covered throughout the semester. There will be a stronger focus on material covered in the latter part of the semester. The evaluation will be in the form of a take-home exam that will be due by 11:59 p.m. on July 12, 2013.

Course Schedule:

You are expected to read the specified chapters in the textbooks and assigned readings before you come to class each week. This will enable you to participate fully in the class discussions and exercises, and gain the maximum benefit out of the class meetings. The schedule of topics and assignments given below is tentative and will be updated as needed.

	Topic	Date	Items Due / Activities
Week 1	SE Chapters 1-3 Introduction to software engineering, development life cycles; Chapters 24, 27 Project management; ACM code of ethics	**May 23 In class**	Exercise: Code of ethics, software practice Project: Form Teams, brainstorm ideas
		May 24 Online	Exercise: Comparative analysis of SDLCs
Week 2	SE Chapters 4-7 Requirements & Analysis; Chapters 8-10 Design; Chapters 11-13 User interface design; patterns; web apps	*May 28 Online*	Quiz 1 Project: Proposal and Specifications
		May 29 Online	Assignment 1 – Tools of the trade wiki pages.
		May 30 In class	Exercise: Requirements and analysis; UML diagrams
		May 31 Online	Exercise: User interface
Week 3	C++ Chapters 2-7 Intro to C++; Chapters 8-9 Functions; Chapter 13 File i/o; secure programming guidelines	*June 3 Online*	Quiz 2
		June 5 Online	Assignment 2 – Testing
		June 6 In class	Exercise: Functions
		June 7 Online	Exercise: Secure programming
Week 4	C++ Chapter 11 Arrays; Chapter 12 Classes; Chapter 16 Overloaded operators	*June 10 Online*	Quiz 3 Project: Analysis Model
		June 12 Online	Assignment 3 – Topics in Software Engineering
		June 13 In class	Exercise: Class definition
		June 14 Online	Exercise: Overloading operators

CSC 260 – Software Engineering
Summer 2013 (May 20 – July 12)

	Topic	Date	Items Due / Activities
Week 5	C++ Chapter 14 Pointers; Chapter 15 Inheritance;	*June 17 Online*	Quiz 4 Project: Design
		June 19 Online	Assignment 4 – Foundational concepts in C++ and classes
		June 20 In class	Exercise: Program with pointers
		June 21 Online	Exercise: Inheritance
Week 6	C++ Chapter 14 Pointers; Chapter 15 Inheritance;	*June 24 Online*	Quiz 5 Project: User Interface Design
		June 26 Online	Assignment 5 – Overloading and Inheritance
		June 27 In class	Exercises: Polymorphism and linked lists
		June 28 Online	Exercise: Linked list class
Week 7	C++ Chapter 14 Pointers and Linked Lists; Chapter 15 Inheritance and Polymorphism;	*July 1 Online*	Quiz 6 Project: Implementation Part I
		July 3 Online	Assignment 6 – Dynamic Memory Management
		July 4 No class	Independence Day Holiday
		July 5 Online	Exercise: Testing
Week 8	C++ Chapters 16-17 templates, STL; Project Presentations	*July 8 Online*	Project: Final Implementation
		July 11 In class	Exercise: Template functions and classes Project: Presentations and Report
		July 12 Online	Final Exam due

Academic Integrity Policy:

Students are bound by the Academic Integrity Policy of the College, available at http://policies.tcnj.edu/policies/digest.php?docId=7642. Unless otherwise stated explicitly, all work for this course, including assignments, quizzes and final evaluation, must be completed individually. Violations will be dealt with in accordance with the policy.

As per the Computer Science Department's policy, a student may not leave the classroom during classroom assessments unless she or he has previously approved documentation that permits her or him to do so.

Electronic Devices in the Classroom:

Students are encouraged to take notes and follow along with the materials posted on the wiki. No other use of electronic devices will be permitted during class time, including listening to music, surfing the Internet, texting, tweeting, Facebook surfing, etc. There will be grade or other penalties for violations of this policy.

Americans with Disabilities Act (ADA) Policy:

A student who has a documented disability and is in need of academic accommodations must notify me immediately. The student is responsible for contacting the Office of Differing Abilities Services (609-771-2571) and ensuring that I receive all required documentation regarding appropriate individualized accommodations in accordance with Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act of 1992. TCNJ's Americans with Disabilities Act (ADA) policy is available at http://policies.tcnj.edu/policies/digest.php?docId=8082.

TCNJ SERVICES

Online Writing Lab: An online writing help system provided by Humanities & Social Sciences Tutoring Services. Students can ask questions and have them answered by a certified writing tutor and gain access to some of the best writing resources available on the web. See: http://www.tcnj.edu/~tutoring/humanities/owl.html.

TCNJ Library: Hours are posted shortly before the semester begins

Instructional Technology Services: A multi-purpose facility designed to assist students and faculty in developing instructional media and other course-related presentational materials. ITS is located in the lower level of the library.