This document is subject to change. I reserve the right to adjust throughout the semester as deemed necessary.

Environmental Note:
In support of our environment, I recommend not printing the syllabus except when absolutely necessary. The syllabus can be accessed from the network drive or stored on your cell phone and/or computer.

Software Engineering (CpSc 488-01) **Spring 2024**

Instructor: Dr. Sam R. Thangiah
Class Time: TR 9:30 – 10:45 a.m. – ATS 224

Office: 250 ATS
Office Hours:
M/W: 10:00 a.m. – 12:000 a.m.
T/Tr: 2:00 - 2:30 p.m. (or by appointment)

E-mail: sam.thangiah@sru.edu
Phone: 724-738-2141

Student Project Meeting Times:
M/W: 12:00 p.m. – 1:30 p.m.
F: 11:00 a.m.. – 12:30 p.m.

URL: http://srufaculty.sru.edu/sam.thangiah
Zoom for weekly project meetings:
https://sru.zoom.us/j/95447703244
Meeting ID: 954 4770 3244
Passcode: 537439

TEXT: Object -Oriented Software Engineering using UML, Patterns and Java, Bernd Bruegge and Allen H. Dutoit, ISBN-13-978-0-13-606125-0
Optional Text: Head First Design Patterns: A Brain-Friendly Guide, O'Reily Media, 2004.

Software Development Platform: Eclipse IDE for Enterprise Java and Web Developers – 2023-12
Java Edition: JDK 17
Other Software: Microsoft Visio for UML diagrams, STS and Window Builder plugin for Eclipse

STORAGE LOCATION (external): \\cpscstorage02.srunet.sruad.edu\java$\CPSC-488-02.0124 or https://cpscjava.sru.edu/CPSC-488-02.0124/

CATALOG DESCRIPTION

The course is the study of software engineering principles and design. The course will emphasize requirement analysis, design, testing, debugging and implementation of a semester long project. Students will work in groups for the project to be implemented. There will be multiple phases for the design and implementation of the project. The project will be implemented using a standard programming environment. Prerequisite: CpSc 374. (3 credits).

GRADING POLICY (approximate points):

2 Examinations	200 points (100 points each)
Final Exam	100 points
Semester Project/Presentation/Weekly Meetings	360 points (210, 30,120)
Assignments	40 points (approximate)
Pop Quiz	80 points approximate (10-20 points each)
Total	780 points (approximate)

TOPICS:

OOP in Java
GitHub
Spring/Maven/Thymeleaf
Chap. 1: Introduction to Software Engineering
Chap. 2: Modeling with UML
Chap. 9: Object Design
Chapter 8: Object Design: Reusing Pattern Solutions
Chap. 4: Requirements Elicitation
Chap. 5: Analysis
Chap. 6: Designing with Use Cases
Chap. 7: Engineering Design Analysis
Chap 11: Testing
Chap 15: Software Life Cycle

COMPUTER LABS:
Artificial Intelligence and Robotics Lab. – ATS 227 – 11 computers + Humanoid, Pepper and Furhat Robots
ATS 129/224/230– 35 computers (can be used when there are no classes)

EXAM DATES:

Exam 1:	Thursday	Feb 22,	9:30-10:45 a.m.
Exam 2:	Tues,	Apr 16,	9:30-10:45 a.m.
Final:	Thursday,	May 2,	8:00 – 10:00 a.m.

General Policies and Procedures for 2023-2024

Attendance Requirements: **Attendance is compulsory. Class attendance will be taken regularly. Lack of attendance usually leads to a failing grade (missed quiz's, assignment, etc.). If you miss a class, it is your responsibility to get the material covered during your absence from a classmate**. For each absence from the class, you will forfeit 10 points from the total score you earn for the class. Absence means not coming to class, coming late to class, leaving the class without the instructor's permission, to mention a few. If you have to miss a class for legitimate reasons, you have to secure my permission for the event in advance and in addition get a letter from a staff or faculty sponsoring the event. If you are absent due to unforeseen medical circumstances, then you need to bring a letter from a medical doctor or a staff at the health center to be excused.

Tardiness: Student's should be in class a few minutes before lectures start. Students late in coming to class will be counted as absent. Each absence will lead to a forfeit of 10 points from your total score for the semester.

Student Conduct and Class Etiquette: Decency and proper etiquette will be observed in class during the lectures and labs. Do not disturb or distract any other students or faculty during the lectures and lab. This includes talking to other students in class. Disregarding proper etiquette will result in removal from the class in addition to disciplinary actions (http://www.sru.edu/offices/student-conduct/code-of-conduct.) **The primary reason that you are attending this class is to acquire knowledge, everything else is secondary.**

When addressing faculty, either by e-mail or other forms of communication, address with the prefix of either Prof. or Dr. followed by their first or last name. Do not address them by their first name unless they have indicated to you that you may. Do not address them as "Yo", "Hey", "Dude", "Hello", "Bro" or any other terms as it is impolite and socially not acceptable.

Assignments/Notes/Data Files from Faculty: Assignments/Notes/Data Files from faculty for each of the classes can be accessed from the cpscstorage02 drive of the SRU campus network. If the class that you are taking is CpSc 100, section 5, then it will be located in /sam.thangiah/cpsc100-5.

Student Folders: Student folders will automatically be created for you in cpscstorage02 drive. The folders will be labeled as \\cpscstorage02\ class>-<section>\<student login>. The student folder should be used for storing all projects, assignments and reports starting from the first week of classes. Individual folders named Projects, Assignments, and Reports should be created to store the respective files. Students not storing the projects, assignments and reports in the folder will forfeit 25 points from the final grade.

In the first week of classes you need to check to ensure that you have a student folder for the class. If you do not have one, you need to send me an e-mail with your name, user id and class and section and request for a folder to be created. If you do not have a folder and do not request for one in the first week of classes, you will lose 25 points when the folder is checked for projects, **assignments** and reports. **It is your responsibility to ensure that you have a student folder for the class.**

Assignments/Reports/Programs from Students: Assignments/Reports/Programs are to be submitted on time at the beginning of the class. Late assignments will receive 0 points. If the class is to start at 1:00 p.m., the assignments must be on the instructor's table before 1:00 p.m. If submitted at or after 1:00 p.m., the assignment will receive 0 points (Synchronize your clock/watch with the U.S. standard time at http://nist.time.gov). No excuses, however creative will be accepted. Softcopies of the assignment/reports are to be submitted in \\cpscstorage02\java$\<class>-<section>\<student login> folder. If you are in the CpSc 110, Section 5 class, it should be placed in \\ cpscstorage02\java$\cpsc110-5\xml5432 where xml5432 is the student login. The simplest way to do this is to create a Map Drive to \\ cpscstorage02\java$ using the My Network Places/Map Network Drive icon on the desktop. Assignments, reports and programs should be placed in folders named Assignments, Reports and Programs respectively. All work submitted for grade should reside in the drive until the final grade is submitted.

Text Books: It is the responsibility of the student to read the textbook that is required for the class. One of the primary reasons that students get lower grades is because they do not read the textbook. The exams and quizzes for the class consist of material from the lecture and from the textbook.

Academic Integrity: Absolutely no instances of academic dishonesty will be tolerated. Plagiarism of another's work or ideas taken from print or electronic media, whether in a research paper or in a computer program, will result in an "F" in the course. Similarly, cheating on an assignment (i.e. copying someone else's work or having someone else do part or entire assignment, doing it as a group), quiz, test, or exam will result in a grade of "F." If one student did the work and a second student copied it, both the students will fail the course.

Academic dishonesty may take many forms. Examples include, but are not limited to, the following:

- Buying, selling, or trading papers, projects, or other assignments.
- Using or attempting to use any unauthorized book, notes, or assistance (for example, copying another student's test or homework).
- Plagiarizing and/or submitting the work of another as your own.
- Completing class work for another person.
- Fabricating information, research results, or citations.
- Facilitating dishonest acts of others pertaining to academic work.
- Conducting research with human subjects without IRB approval.
- Conducting research with animals without IACUC approval.
- Possessing or sharing unauthorized examinations.
- Submitting, without instructor permission, work previously used.
- Tampering with the academic work of another person.
- Recording an instructor's teaching content and/or distributing a professor's teaching content without permission.
- Ghost-taking an exam in place of a student or having any person take an exam in your place.
- Any attempt to falsify an assigned grade on an examination, report, or program or in a grade book, document, or other record.
- Any attempted, or actual computer program theft, illegal use of software; illegal downloading or streaming of copyrighted media, or inappropriate use of the Internet; such as, but not limited to, illegal or unauthorized transmission; or improper access to any computer system or account.
- Any attempted, or actual, collusion willfully giving or receiving unauthorized or unacknowledged assistance on any assignment or examination (all parties are considered responsible).
- Forging a faculty member's or administrator's signature on any document.
- Copying and pasting digital media (including but not limited to, email correspondence, text, images, or other media from online sources) without proper citation, the copyright owner's permission to use the digital media, or evidence of having performed a favorable fair use analysis.
- Copying and pasting significant portions of digital media with or without citation.
- Submitting work generated by artificial intelligence as your own.

In addition, if you did an assignment/exam and do not know how to replicate it, it will be considered as cheating/plagiarism. Faculty have additional access to the electronic reporting system to report academic dishonesty for students and these reports are permanent. Furthermore, any academic dishonesty can result in probation/suspension/expulsion from the University. Check the following document on the university policy on academic dishonesty (https://catalog.sru.edu/academic-policies/academic-integrity/)

Use of Scheduled PC Lab Time: Classes scheduled for the PC Lab are to be used for working on assignments for the course in which you are enrolled. They are not to be used for sending and receiving e-mail or using messenger, or for "surfing" the Internet. Any student who violates this policy will be asked to leave the PC Lab, and may find him/herself failing the course in which he/she is enrolled. **You are not to leave the lab. even if you complete the required exercises during the class period. Leaving the lab. before the class period is complete is considered as an absence. If students do not attend the lab. for classes that meet once a week, it will also be considered as an absence. Use of the computer in the lab. during the class period to do anything that is not related to class will result in the loss of 10 points for each occurrence.**

Incomplete Grades: An “I” may be granted when extenuating circumstances prevent completion of course requirements within the course’s time period, and with the mutual understanding that the final grade will be one letter grade less than the grade the student would have received if the work had been completed on time. An “I” is converted to an “F” if it is still unresolved by the end of the succeeding semester. An “I” grade will be awarded in extreme rare cases.

Cell Phones/MP3 Players/Head Phones/Electronic Devices: All Cell phones, mp3 players, watch alarms, head phones, Ipods (any devices with earphones), watches synched to phones, and electronic devices should be **turned off** and placed away from audio/visual/physical range during the lectures and lab sessions. Contact me before class if the phone needs to be active to receive emergency calls. **Use of cell phones or any electronic devices in class will result in the loss of 10 points for each usage**. **A cell phone within visual range to see incoming calls or texts will result in loss of 10 points for each occurrence**. In short, put the cell phone away and forget about its existence for the length of the class period.

Withdrawal from Class: A student can withdraw during the first ten weeks of class with a “W”. The “W” is a normal withdrawal that will not affect the students QPA. After the first ten weeks, no withdrawals are allowed and the student will receive a letter grade. That is, after the 10th week, you cannot drop out of the class.

Extra Credit: There will be no extra credit assignments/projects/programs assigned at any point in the semester.

Semester/Final Exams: There will be no rescheduling of semester, mid-term and final exams. Students are expected to take the exams respective to the scheduled times.

Demonstration of Computer Programs in class
When computer programs are demonstrated in class, they are running on a university system with standard settings for the Windows operating system. If you are unable to run the computer programs demonstrated in class on your personal computers, I will not be able to help you with the issues as each personal computer system can have its own unique settings. I can only show you how the programs being demonstrated run on a university system, namely one that is in a class or lab. associated with the computer science department.

E-mail: All students are expected to have an e-mail address. E-mails should be checked on a regular basis for notes/assignments from the faculty that might be due in the succeeding class period. Students can access their e-mail from any computer using a browser and the address http://sruwebaccess.sru.edu. When sending an e-mail to me, the following information should be at the top of the e-mail. Any e-mail that does not have the following information will be discarded.

First Name Last Name
User Id
CpSc <class number>>- <<class section>>

Grading Scale:	A	90 – 100%
	B	80 – 89.99%
	C	70 – 79.99%
	D	60 – 69.99%
	F	0 – 59.99%

I will try and respond to e-mails within 24 to 48 hours. All e-mail response will be during work hours M-F 8:30 – 4:30 p.m.

Projects in Software Engineering

- Students will be placed in groups of two or three and will be given projects that would require them to work as a group in accomplishing the task. Group project means you will work on your sub-part of the code, as will each of the other individuals in the group, towards getting the entire project to work. The faculty reserves the right to split the project into sub-parts and have students work on the separate sub-parts. **If progress is not being made in the project by any individual/individuals in the group, the faculty reserves the right to split the group up in any manner or form including the assignment of new projects to an individual or individuals in the group**.
- When sending me e-mails related to group projects, in the first line of the e-mail include the group number and group name so that I have a reference to which group I am communicating with.
- All projects should be in the Github project account, https://github.com. Each student will have created a Github account with your SRU userid (this needs to be done before the first class of the semester). The students e-mail address has to be the SRU e-mail address for the Github account. For example, if your e-mail address is abc1234@sru.edu the Github userid should be abc1234). The faculty will create a Project for each of the groups. All code should be updated to this account. This will allow the faculty to track the progress being made by each student with respect to the project.
- Ensure your Github account has your full name for me to identify the id with the name.
- For semester long projects, every Tuesday before the start of the class, or before the weekly faculty meeting if it is earlier than Tuesday, a report on the progress of the project should be submitted in softcopy format. The softcopy should be placed in the cpscstorage02 drive. Late softcopy reports will result in a 0 for that report. The reports should remain in the cpscstorage02 for the entire semester.
- The weekly meetings will be for students doing the semester long project. If the weekly meeting is on a day other than a Tuesday, a second individual and group report should be submitted on what was done from Tuesday till the day of the meeting.
- Each individual will have an individual folder in cpscstorage02. Each group will have a group folder in cpscstorage02 with all students in a group having access to the group folder.
- Each individual folder should consist of a Report, Assignment and other folders as needed. Each group folder should consist of a Report, Final and other folders as needed. The group Report folder is for weekly group reports and the Final folder is for submission of the final project including the code and the documents. The group report should be what all the individuals of the group have achieved as a group for the week.
- The group Report folder is for weekly group reports on the contributions made by the group and will be in a folder named Reports in the group folder.
-

- The individual Report folder is for weekly individual reports on the contributions made by an individual and will be named Reports in the individual's folder.
- For weekly meetings, you will get credit for the amount of work that you have put into progressing towards the project being completed. The work needs to be MERGED into the master. Just writing the code and submitting it to Github gets you no credit as it says nothing about the code or whether it even works. Or updating the documents and writing a few lines of code gets you no credit. All this does is makes the others in the team work on the heavier load. If one or more are taking a heavier load, they will get more credit than those who do less work, no exception.
- Within a day of the weekly meeting, each group should place a task report outlining the tasks that are to be completed by each of the individuals in the group for the next week. These tasks should be listed in the Issues section of the Github project and the groups members will check out the tasks and complete each of the tasks listed. The task file should be named Task Report-<Week #)-<mm/dd/yyyy of the report>.
- Each member of the team must create Issues in the Github on the part of the project that they are working on and check the Issue out to themselves. Then they will code to address the issue, push it to the Github, merge the code and then close the issue. Issues in Github are like logs that is indicated which group member took responsibility to do what and if they completed what they said they were going to complete. Not creating Issues when contributing towards the project in Github will result in a 0 for the weekly meetings and loss of points for the entire project.
- All updates to projects in Github should be completed and updated/merged before 7:00 p.m. on the day before the group meeting. You will be graded on what is on Github at 7:00 p.m. or before on the day before the group meeting. There is no exception to this rule as it takes me an hour or more to go through each of the projects to see what has been done by each group and each individual in the group.
- All reports, Group and individual, weekly, group and weekly task, should be of the format Weekly Report-<Week #)-<mm/dd/yyyy of the report>
- Each report should have the name of the participant(s), the name of the project, the date that it is being handed in and the details of the project. Each report should show progress towards the completion of the project. A softcopy of the form that is to be used for submitting the report is available in the cpscstorage02 drive and is also attached to this syllabus.
- For semester long project, every week, during times designated within the office hours, each group will meet with the faculty for 15 to 20 minutes and explain what has been accomplished. The faculty will discuss with each student the student accomplishments for the week and progress towards the completion of the project.
- The points assigned to each individual for the weekly report will be between 0 and 10 points and will be based on the individual contribution of an individual towards the project, the group report, the individual report and attendance of the weekly meeting.
- Ensure that the project does not have compilation errors, is working and there is clear documentation on how I have to install it and what logins and password I have to use, including the group reports and individual reports.
- Ensure that there is an install manual for me to download and run your code. The install manual should also have all the login account information. The install manual should be updated before 7 p.m. the day before the meeting.

- Ensure there is a Reference document before each meeting. The Reference document lists all external resources that you accessed or used for the project. External resources include articles, codes, web sites and other resources used or accessed. Not listing external resources or using resources obtained externally and not listing it will be considered as plagiarism. The References document should be updated weekly and before 7 p.m. the day before the meeting.
- Ensure your project can be run by me on my office system. To make that happen, you need to use a computer that is in ATS 129, 224 230 or 232, download the project from Github into the required Eclipse version and then run it. Just because the program runs on your system does not mean I will be able to run it on my office computer.
- Any project that is submitted with compilation errors will get an automatic 0 for the weekly meetings.
- For semester long project, missing a meeting or failing to hand in a report on a Tuesday will result in 0 points towards the report for that week. If you miss more than one meeting without an acceptable excuse, you will be dropped from the group and will get a 0 towards the semester project. If an entire group misses more than one meeting without an acceptable excuse, your weekly meeting time will be cancelled for the entire semester. You will need to contact the faculty to be rescheduled for weekly meetings.
- The faculty will check and monitor the Github for contributions being made by each student in a group. Each student should be individually ready to answer questions with respect to the project and the code.
- On the Monday of the twelfth week of classes 99% of required code should have been completed.
- On the Monday of the thirteenth week of classes 99% of the manuals and presentation should have been complete and before 5:00 p.m. on Monday all code, document and presentations, in PowerPoint slide format, should have been submitted to the Final folder in the group folders.
- On the Tuesday and Thursday of the fourteenth week of classes, presentations will be made to the class using PowerPoint.
- Before the end of the fourteenth week of classes, I will provide a grading for the final project and any corrections that need to be done for the final project. The group projects should have all the corrections made and submitted to Github and cpscstorage02 before the final exam day.

The reports are due staring from the second week of classes

CpSc 488: Software Engineering
Weekly Report Template
Dr. Sam R. Thangiah Fall 2023
(10 points)

Revision Date: January 26, 2024 Weekly Report
(The weekly report should be typed. Handwritten reports will not be accepted)

Date of Report:
Title of Project:
Group Number:

1 -Your Name:
2 - Partner’s Name:
3 - Partner’s Name:

Percent Work Contributed (the total should not exceed 100%)

Group Member	Percent of Work done
1 - Self	
2	
3	
4	
5	

Number of group meetings for the week:
Number of total hours spent on group meetings:

For Official Use Only:
Total points Possible: 5 Total Points Received:______
Attended meeting with Faculty: □

“All the information that has been reported is true to the best of my knowledge. I attest to the truthfulness of the report by signing below“

Signature:______________________________

Goal from previous meeting:

Goals achieved:

Goals not achieved:

Goal for next meeting:

Check if the following has been accomplished:

□ Code is documented
□ Code has been updated to GitHub

CpSc 488: Software Engineering Dr. Sam R. Thangiah Spring 2024
Revision Date: January 25, 2024 Weekly Tasks

(Within one day of the weekly, meeting the following form should be placed in the group folder for each week indicating the tasks that are to be completed by each group member.

These tasks should map with the issues in the Github project)

1 - Group Member 1:
2 - Group Member 2:
3 - Group Member 3:
4 - Group Member 4:

Tasks to be completed for next week by group members:

Group Member 1:

Group Member 2:

Group Member 3:

Group Member 4:

CpSc 488: Final Project Grading (subject to change)

Software Engineering

Instructor: Dr. Sam R. Thangiah **Fall 2023**

Percent of Work Completed (PWC): _____%

Deadline for Submission: <TBD>

Final Project Evaluation (100 points):

Completion of program	**140 points:**	
Code Executes Completely		_____
Code Reusability		_____
Code Testing/Logging		_____
Check for Exceptions		_____
User Interface		_____
If incomplete, explain all details of it		_____
Documentation	**70 points**	_____
Glossary/External Sources		_____
Problem/Statement /Requirements Document		_____
UML/DataFlow/ER/Other Diagrams		_____
Documentation in Code		_____
Technical Manual		_____
Security Manual		_____
User Manual		_____
Evaluation Manual		_____
Install/Config/Login Manual		_____
Weekly Meetings	**120 points**	_____
Presentation	**30 points**	_____
Bonus Points	_________	_____
Total	***360 points***	_____

CpSc 488: Software Engineering

Instructor: Dr. Sam R. Thangiah

Student Name:

Percent of Work Completed (PWC): ____%

Final Project Presentation (30 points):

Availability of slides before presentation (2 pts.) _____

Structure (8 pts.)

(title, intro. motivation, body,

summary, future work _____

Delivery (5 pts.) _____

Coverage and Knowledge of Content (5 pts.) _____

Quality of Content (5 pts.) _____

Engagement with Audience (5 pts.) _____

Bonus Points _________ _____

Total ***30 points * PWC*** ____

The final project reports for the Software Engineering class will be based on the following criteria:

- **User Manual:** A user manual is for a user that wants to use the program and not make any changes to the programs. The user manual should not be a technical document.
- **Technical Manual:** A technical manual is for a programmer that wants to understand the program and make any changes to the programs. The technical manual should have all the technical details required to understand the program. Material from the final report can be used towards the technical manual.
- **Evaluation Manual:** An evaluation manual will consist of three categories. Namely: 1) Problems – List the current problems with the systems, 2) Cleanup – How the problems can be fixed and 3) Improvement/Expansion – How the program can be improved and expanded.
- **Installation Manual/Config Manual –** The set of procedures that needs to be done to install the program and execute it.
- **Security Manual: –** Detailed explanation of what can go wrong and what did you do about it
- **Login Manual: –** All logins that are to be used in the program should be listed in this manual.
- **Hardware/Software Requirement:** The minimum hardware and software requirements that would be needed to execute the program.
- **Final Report:**
    - **Header Page** – The first page of the report should be the header page with the title of the project and the names of the group members. Next to each group member's name, put the login name of the group member (e.g. Sam R. Thangiah (srt1234@sru.edu).
    - **Reference:** Full reference of the any papers used for implementing the system/algorithm (see the reference section of the paper on how to quote a reference).
    - **Completion:** Complete/Incomplete. List the program as complete if all the algorithms in the paper were completed. (A project is complete if the modules that were skipped were done with my permission). List all the algorithms and modules in separate lines that were implemented in the program.
    - **Contribution:** Explain what percent of the code was available from different sources (commercial, freeware, open-source, or other and document the locations from which they were obtained) and what percent was contributed by your group.
    - **External Sources**
        - List all external sources used for the project. Not listing/or incomplete listing of external sources will result in an automatic 0 for the project and a failing grade. It is plagiarism to use external sources and not list the source.
        - External sources are any source that was access or used and includes:
            - Code (list the link from where code was obtained)
            - Package (list the link to the package)
            - Documents with or without code (list the link to the document with or without the code)
            - Document without code ((list the link to the document with or without the code)

        - Book (List the full information on the book)
            - Person (list the name and contact for the person who helped you with **the coding)**
    - If you used/cloned/copied code from a web site, or got help from anyone or any site to do the coding, and did not list it in the References section, it will be considered as plagiarism and you will get an automatic 0 for the project and an F for the class.
- **Glossary**: A list of technical works used in the domain together with their meaning. Application domain is the target area in which the program is to be used.
- **Problem Explanation**: A detailed explanation of the problem that is being solved.
- **System Requirements –**The System Requirements document that was used for implementing the project
- **UML Diagrams** – UML diagrams associated with the complex part of system requirements
- **Data Flow Diagrams** – Data Flow diagrams to explain the working of the complex part of the system.
- **Entity/Relationship Diagrams** – Critical if databases were implemented in the program and their relationship are to be explained.
- Other diagrams that helps in the understanding of the requirements.
- **Caveats/Minefields**: Include all the caveats that a layman should know about when attempting this project and all the minefields that should be sidestepped.
- **Documentation –** The program should be fully documented. The documentation should be in the hierarchy of File, Class and Method. The File gives an overall view, the Class gives a detailed view and the Method gives a fine view. The internal document should be done in a manner such that the javadoc programming tool can be used to generate the standard Java Document pages. For more information, see **http://java.sun.com/j2se/javadoc/**.
- **File Path Names** - Place all path names for files being used in the program in an OO class with variables defined to be private.
- **Code Reusability –** Was the program (Classes and Methods) designed in a manner to allow code reusability. Justify and explain in detail.
- **Testing** –
    - Human-Computer Testing
    - Document on Blackbox testing
    - Document on Whitebox testing
    - Document on Unit testing
    - Document on Integration testing
    - Document on Regression testing (Indicate why certain test were selected to be part of the Regression testing)
    - Document on Boundary Analysis
    - Document on Volume Test
    - Document on Stress Test
    - Document on Missing Resources (files or databases)
    - Document on Mis-Management of Resources (locked files or databases that are not released)
    - Document on Recovering from Crashes (Kill the program unexpectedly and bring it up)
    - At least 10 critical test cases must be conducted and documented.

    If testing fails and it was not fixed in the program, indicate the ones that failed and why there were not fixed (Do not wait for me to find them as it will lead to loss of points)
    - **Logger –** Use of the logger class to do both testing and also to analyze the working of the program.
    - **Deployment/Maintenance –** Document on deploying the program and issues with maintenance.
    - **Post-Mortem Analysis** – Once the project has been completed, look back it and explain what you would have done differently and why. (You would be kidding me if you said you did a perfect job, so don't even try it!!!)
- **Grammatical and Spelling Errors -** The report should be free of grammatical and spelling errors. Read and read the paper again. Even though you might have good results, the strength of your work is on the written report. A poorly written report will result in loss of points.
- **Working Copy of Program in Designated Folder –** A working copy of the program and all documentation should be placed in the cpscstorage02 folder named Final and also in GitHub. All students should have the Final project folder in the cpscstorage02 directory. That is, if there are two members in a group both members should have a copy of the final project in their respective directory. There should be a Code and Manual folder and it should consist of the code and manuals.

Computer Science Department
Course Competency Plan

Catalog Description

The course is the study of software engineering principles and design. The course will emphasize requirement analysis, design, testing, debugging and implementation of a semester long project. Students will work in groups for the project to be implemented. There will be multiple phases for the design and implementation of the project. The project will be implemented using a standard programming environment. Prerequisite: CpSc 374. (3 credits).

Course Outcomes

This course and its outcomes support the Computer Science Learning Outcomes of **Problem Solving and Critical Thinking** (PS&CT), **Communication and Interpersonal Skills** (C&IS) and **Ethical and Professional Responsibilities** (E&PR). These Computer Science Learning Outcomes are tied directly to the University Wide Outcomes of **Critical Thinking and Problem Solving, Communication**, and **Values and Ethics**.

This course engages students in the following high impact practice: writing-intensive

Program Learning Outcomes	Course Objectives
C & IS a. Document all aspects of a system precisely and clearly	1. Document all aspects of software engineering: analysis, design, build and test.
C & IS b. Use written, oral, and electronic communication to convey technical information effectively	2. Write a design document, user manual and technical manual.
	3. Give an oral presentation on software engineering topic
C & IS c. Devise effective user interfaces based on the application	4. Interface testing of code with heterogeneous groups
C & IS d. Work cooperatively in teams and with others	5. Participate in a semester long team project on analyzing, designing, building, testing and deploying software.
E & PR d. Plan for and ensure the security, privacy, and integrity of data	6. Detail privacy and security issues in the development of software
E & PR c. Demonstrate an awareness of the codes of professional ethics in the information technology industry	7. Identify the ethical and professional responsibilities of a software engineer
E & PR e. Recognize the need for continuing professional development	8. Detail new trends in software engineering

Additional Course Objectives include:
The student will be able to:

1. Identify the necessary steps needed to write a requirements analysis.
2. Identify the necessary steps needed to write a domain analysis for a given requirement analysis.
3. Identify methods for coding a prototype for a given design document.
4. Identify the necessary steps needed to code, test, and debug a given design document.

Copyright Statement - Students shall adhere to the laws governing the use of copyrighted materials. They must ensure that their activities comply with fair use and in no way infringe on the copyright or other proprietary rights of others. Additional information regarding copyright can be found here http://www.copyright.gov/ and information about fair use can be found here http://www.copyright.gov/help/faq/faq-fairuse.html.

Reporting Obligations of Faculty Members under Title IX of the Education Amendments of 1972

Slippery Rock University and its faculty are committed to assuring a safe and productive educational environment for all students. In order to comply with the requirements of Title IX of the Education Amendments of 1972 and the University's commitment to offering supportive measures in accordance with the new regulations issued under Title IX, the University requires faculty members to report incidents of sexual violence shared by students to the University's Title IX Coordinator. The only exceptions to the faculty member's reporting obligation are when incidents of sexual violence are communicated by a student during a classroom discussion, in a writing assignment for a class, or as part of a University-approved research project. Faculty members are obligated to report sexual violence or any other abuse of a student who was, or is, a child (a person under 18 years of age) when the abuse allegedly occurred to the person designated in the University protection of minors policy. Information regarding the reporting of sexual violence and the resources that are available to victims of sexual violence is set forth at: https://www.sru.edu/offices/human-resources-and-compliance/sexual-misconduct-and-title-ix-resources.

No-discrimination: Slippery Rock University of Pennsylvania does not discriminate on the basis of race, color, sex, sexual orientation, gender identity, gender expression, national origin, religion, age, disability, or veteran status in its programs or activities in accordance with Title IX of the Educational Amendments of 1972, the Americans with Disabilities Act of 1990, Section 504 of the Rehabilitation Act of 1973, Title VII of the Civil Rights Act of 1964, and other applicable statutes and University policies. https://www.sru.edu/offices/human-resources-and-compliance/notice-of-non-discrimination

Correct Pronouns, Names, and Inclusive

Correct Pronouns, Names, and Inclusion: Language is gender-inclusive and non-sexist when we use words that affirm and respect. I hope to create a space where students have the opportunity to bring all aspects of themselves into the classroom. I support people of all gender expressions and gender identities, and I encourage students to use the name and set of pronouns which best reflect who they are. In this spirit, I expect all students to use the correct name and pronouns of their classmates. I will respect and use the language you use to refer to yourself, and I will encourage other members of our classroom community to do the same. Please inform me if my documentation reflects a name or set of pronouns different from what you use. I welcome and will respect if at any point you wish to update your name or pronouns with me and/or with the class. If you have any questions or concerns, please contact me after class, by email, or during office hours.