TORRENS UNIVERSITY AUSTRALIA

ASSESSMENT 3 BRIEF	
Subject Code and Title	SEP401 Software Engineering Principles
Assessment	Final Project – **Software Application Development** and **Presentation**
Individual/Group	Individual/Collaborative
Length	Presentation Duration – maximum of 12 minutes
Learning Outcomes	This assessment addresses the Subject Learning Outcomes outlined at the bottom of this document.
Submission	Face to Face and Online Students 6-week delivery: Due by 11:55pm AEST/AEDT, Wednesday of Week 6 12-week delivery: Due by 11:55pm AEST/AEDT, Wednesday of Week 12
Weighting	35%
Total Marks	100 marks

Context

Your final project focuses on the software application development and presentation. The software application development assesses your programming skills, ability to translate the design document into a code, ensure quality of this code and test them while the presentation assesses your ability to communicate your ideas to stakeholders.

As a software developer (or programmer), it is important to have deep technical expertise and hands-on experience on key programming languages. Software developers are expected to understand design and be able to translate them into code.

Developing your presentation skills as a software engineer will help you convey your ideas to your team and stakeholders.

Instructions

1. Create a test plan based on your revised Software Requirements Specification document.
    a. Specify following details for each test case:
        i. Identification number
        ii. Descriptive title
        iii. Instructions
    b. Assign appropriate severity to each test case:
        i. Critical (Test cases critical to the success of software)
        ii. Important (Test cases encountered on day to day functional tasks)
        iii. Workaround (Test cases for which the software could run even with the defect)
2. Develop your software application using C++ based on your revised Software Design Specification document. You can use third party libraries.
3. Make sure that your software is free of errors and programming bugs.
4. Prepare and record your presentation (maximum 12 minutes) ensuring you have included:
    a. An overview of your project and its objectives
    b. A clear demonstration of the functionalities of your software.
    c. Discussion on how you tested your software using your test plan and what the result was.

Submission Instructions

1. The following 4 items should be **zipped** and submitted via Assessment link in the main navigation menu in SEP104 Software Engineering Principles:
    a. Source codes and applicable libraries.
    b. Test Plan
    c. Digital Presentation slides
    d. Recording of your oral presentations (eg. Screen recording with audio using Quick Time Player or other suitable program) and save the recording as a .MP4 file for submission.

Important: Ensure each item has been saved with the following file name convention: **subjectcode_first initial_surname_assessment number and part**. E.g., **SEP401_J_Smith_A3_a**

The Learning Facilitator will provide feedback after your presentation and also via the Grade Centre in the LMS portal. Feedback can be viewed via My Grades.

Assessment Rubric

Assessment Attributes	Fail (Unacceptable) 0-49%	Pass (Functional) 50-64%	Credit (Proficient) 65-74%	Distinction (Advanced) 75 -84%	High Distinction (Exceptional) 85-100%
Software Application meets project objectives ***20%***	Very little if any of the project objectives (functional and non-functional) appear to be met in any clear way. The program did not work or had many bugs	Some of the project objectives (functional and non-functional) were met. The program produced some correct results and/or had several bugs.	The project objectives (functional and non-functional) were met but clearly not in entirety and/or there were some clear deficiencies. The program produced correct results for most project specification and/or had few bugs.	The project objectives (functional and non-functional) were met with few cases that may be questionable and/or additional aspects were needed to be explored. The program produced the correct results and displayed them correctly for almost all project specifications with no bugs.	The project objects (functional and non-functional) have all been addressed and showed all aspects have been explored. The program met all of the project specifications with no bugs.
Usability ***10%***	Software was not intuitive to use, there was little or no instructions, excessive effort was required to use the software. The program did not display results correctly and did not meet most display specifications.	Some parts of the software were intuitive to use but still needs a lot of effort. In most cases, there were some aspects of using the software that were confusing even with some instructions. The program did not display results clearly or did not meet most display specifications.	The software can be used with some effort. Very few cases that were confusing on how to use the software. The program displayed results clearly but some cases but there are some aspects that needed some effort.	The software can be used intuitively with little effort. Very few unexpected outcomes. The program displayed results clearly and met most of the display	The software is easy to use, intuitive and outcomes are as expected in all cases.

				specifications.	
Application of programming knowledge ***20%***	Demonstrated no or little ability to make use of appropriate third-party libraries or systems. Demonstrated no or little programming skills. Inappropriate programming approaches were used frequently.	Used some third-party libraries or system appropriately. Showed some basic programming skills but some inappropriate programming approaches were used.	Reasonable use of some third-party libraries or systems but lacked the capacity to make use of them and/or some choices could have been better in hindsight. Showed reasonable demonstration of programming skills and knowledge in terms of data structures, algorithms and other applicable theory, but only in some parts. Lacked sophistication in programming.	Appropriate use of some third-party libraries or systems with very little choices could have been better in hindsight. Showed reasonable demonstration of programming skills and knowledge in terms of data structures, algorithms and other applicable theory in most parts and showed some sophistication in programming skills.	Excellent choices of third-party libraries or systems and clearly established ability to use them with few if any limitations. Consistently showed application of programming skills and knowledge in terms of data structures, algorithms and other applicable theory. Where appropriate, showed sophistication and a high-level appreciation of programming intricacies.
Quality of source code and Error Handling ***15%***	Extremely hard to read, little or no attention to code formatting, lack of logical organisation, little or no appreciation of coding and commenting practices. The program did not check error conditions.	Code is somewhat readable but needed a lot of effort to understand. Some formatting were applied but not consistent. Very little comments were provided. The program checked for few error conditions and did handle them appropriately.	Reasonable clarity and code can be followed with some effort. Comments helped to understand the code. Logical organization was lacking in some ways and there were a number of questionable expressions. The program checked for some error conditions and handles them appropriately	Good standard of coding and commenting were evident. Code was easy to read with some expressions needing improvement. Commenting was evident and code was somewhat easy to maintain. The program checked for most error conditions and handled them appropriately.	High standard of coding and commenting was consistently applied. The code was easy to read and maintain. The program checked for all error conditions and handled them appropriately.

Test plan ***25%***	Test cases are absent or very few, and are poorly documented or undocumented ; bugs not documented	Test cases miss significant scenarios, and are poorly documented; bugs are poorly documented but mostly incomplete and some are incorrect.	Tests cover most representative cases, tests and known bugs are adequately documented	Test cases are thorough and systematic, known bugs are documented	Test cases are thorough and systematic, well documented with expected and actual output
Oral presentation ***10%***	Uses eye contact ineffectively as entire report is read from notes Does not show any form or organisation and preparation. Fails to speak clearly and audibly and uses unsuitable pace. Does not engage audience. Does not have grasp of information and cannot answer questions about subject Does not clearly define subject and purpose; provides weak or no support of subject; gives insufficient support for	Some eye contact, while reading mostly from notes. Speaks in uneven volume with little or no inflection. Occasionally engages audience. Shows some form of organisation and preparation but uncomfortable with information and is able to answer only rudimentary questions. Attempts to define purpose and subject; provides weak examples, which do not adequately support the subject.	Maintains eye contact, glances at the notes most of the time. Speaks clearly and uses suitable volume and pace. Takes steps to engage the audience. Shows organisation and preparation. Is at ease with expected answers to all questions without elaboration. Has somewhat clear purpose and subject; some examples that support the subject.	Effectively uses eye contact, glances at the notes some of the time. Speaks clearly, effectively and confidently using suitable volume and pace. Fully engages the audience. Shows effective organisation and preparation: introduces the topic clearly and somewhat creatively. Maintains clear focus on the topic. Demonstrates knowledge by answering class questions with some explanations. Provides clear purpose and subject: examples and facts; support conclusions with evidence.	Highly effective uses eye contact. Holds attention of entire audience with used of direct eye contact, seldom looking at notes. Speaks exceptionally clearly, effectively and confidently using suitable volume and pace. Fully engages the audience. Shows exceptional organisation and preparation: introduces the topic clearly and creatively. Constantly maintains clear focus on the topic and effectively includes smooth transitions to connect key points. Demonstrates full knowledge by answering all

	ideas or conclusions				audience questions with explanation and elaboration. Provides clear purpose and subject: pertinent examples and/or facts; support conclusions with detailed evidence.
The following Subject Learning Outcomes are addressed in this assessment					
SLO c)	Design, develop, maintain and evaluate software systems.				
SLO d)	Produce high quality software solutions.				
SLO e)	Meet the core ethical standards of the software development process and legal responsibilities.				