Winter 2021
First Year Competition

PROGRAMMING

Competitor Information Package

WATERLOO
ENGINEERING SOCIETY

Winter 2021 First Year Competitor Schedule

Time	Activity	Platform
09:00 - 09:15	Registration and Verification Complete	Discord
09:15 - 09:30	General Welcome Presentation	Discord #Opening Ceremony
09:30 - 10:00	Programming Competition Introduction and Questions	Discord #Competition Introduction *Please join the programming competition voice channel*
10:00 - 16:00	Design and Build	Self - Administered
16:00 - 16:30	Break	N/A
16:30 - 19:00	Presentations and Judging	Discord #Programming Judging Room
19:30 - 20:00	Judges Decision and Awards	Discord #Closing Ceremonies

Times are approximate. Exam event timing may change on the day of the competition, which will be communicated in the* *#Announcements*** *Channel on the Discord Server*

General Rules & Guidelines

The following rules and guidelines are to be followed by teams, as the competition is hosted online:

1. After the initial welcome ceremony, the competitors will split off into their respectives competitions and be presented with their problem spaces and design challenges. Please try to ask all questions during the question period following the reveal of the challenge so that you have the full attention of the organizers. Beyond the questions period, you may ask questions in the Discord Server in the Programming **#Help Desk** channel
2. Competition - related questions will only be answered in the Programming **#Help Desk** channel to ensure fairness. If the question asked affect all participants, an announcement will be made in the **#Announcements** channel. Teams are expected to monitor the Discord Server for updates about the event and potential schedule changes.
3. For personal emergencies or sensitive questions, please contact one of the Competition Directors directly: **Stuti Munshi or Yi Nan Zhang**
4. Project submissions must be original solutions conducted by the competitors during the allocated competition times with all sources correctly cited.
5. Time remaining in the Design & Build stage will be announced to competitors at the 3-hour, 1-hour, 30-minute and 10-minute marks
6. Teams are not allowed to make any changes to their submission after the design stage is over. Dishonesty will not be tolerated and will result in disqualification. The final version of the project and presentation must be submitted to this **form** at the end of the Design & Build stage.
7. If the discovery of a rule violation occurs during the competition, immediately following the competition, competitors have 24 hours to initiate an appeal for the decision. If the discovery of a rule violation occurs after the competition, the competitors will be immediately notified via email and will have three (3) days to appeal the decision. Should a competitor fall into one of these situations, the competitor should **contact the First-Year Commissioner, Alaina Hansen**

Virtual Competition Information & Guidelines

The following rules and guidelines are to be followed by teams, as the competition is hosted online this term:

1. Teams are expected to have an open line of communication with the Competition Directors and respond to messages in a timely manner throughout the duration of the event.
2. The Welcome and Competition Introductions shall be hosted via a Discord Voice Channel. The joining information will be emailed to the competitors and posted in the Discord #**Welcome** Channel.
3. During the Design stage, teams may use any video or voice conferencing platform that are comfortable with. Please ensure that the platform supports screen sharing to promote collaboration. The organizers will be active on Discord for any support or questions needed. If a voice/video call with the Director is required, the Director may join the team's ongoing call, or they can set up a new call on a platform of their choosing.
4. Judging Schedules will be released at 16:00, with the first presentation slot occurring at 16:30.
5. During the Presentations & Judging stage, the judges will be in a Discord call in their respective voice channels for the duration of the stage. Teams should join the **#Waiting Room** 5 minutes before their time slot. Teams will be given direction on when to entire their competitions judging room. Teams will present their own slides during their slot; please ensure at least one team member is able to share their screen during the presentation. In case there are technical difficulties the judges will also have copies of the slides. When you are in the waiting room, a volunteer will take the attendance of all team members to ensure everyone is present.
6. After the judges question period, the presentation team shall leave, and judges will be given a 5-minute consideration period before the next team presents.
7. After all teams have presented, the judges will be given 10 minutes to deliberate among themselves to compare evaluations and determine winners. Teams should be ready to join the **#Closing Ceremony** for the judges' feedback and announcement of winners after all teams finish presenting.

Additional Programming Competition Guidelines:

1. **Plagiarism** and **cheating** will not be tolerated. Use of third-party software libraries is permitted and encourages if third-party code is acknowledged and all licensing requirements are compiled with. Any attempt by competitors to claim third-party code as their own is considered plagiarism and is grounds for disqualification.
2. Competitors *may use any language* or platform to complete the competition. However, note that the final product must have a way to produce an appropriate output file that can be viewed by the judges, as detailed later in this package.
3. The submitted product should contain thorough instructions on how to build and run the program, including instructions for installing any third-party dependencies.
4. The final product and presentation materials should be submitted via Microsoft Forms to the Please ensure at least one team member is able to share their screen during the **form** submitted at the end of the competition. Please submit source code as a *text file* to the **form**.

Scenario/Problem

Poker is a game that requires a skill to be developed over a period of time in order to be considered a “Professional poker player.” For many of us, we see the game as a leisure activity that is played amongst our friends. One day, a professor from the Engineering Department overheard their students discussing the game of poker and was intrigued by their conversation. The professor challenged their students to a game of poker on Easter Weekend. However, little did the professor know that all his students were proficient in the game of poker, except for themselves. The last thing they wanted was to embarrass themselves in front of their own students!

Objective

Prior to the professor’s match against their students, they wanted to study the game a little bit, but not just by reading up on different plays; they wanted to get a little creative. You will be developing a program that simulates the game so the professor can see what it is like to play a sample game of poker. There is more than one poker hand that is attainable, however, some are stronger compared to others.

Details

The game of poker can be complicated to understand, however, once the fundamentals are understood, it becomes quite simple.

For the purposes of this challenge, assume there is only one player (ie. the program does not to be implemented for multiple players)

Before you start reading through the details, we advise that you watch this short *YouTube tutorial* on how the different hands in poker work, if you are unfamiliar with the game!

Link to Tutorial: https://www.youtube.com/watch?v=w4Kj6Ra_BTw

In the game of poker, there is a deck of 52 playing cards, excluding jokers. Before the deck is distributed, it must be shuffled thoroughly to ensure the game is fair and randomized. The program should be able to draw two cards initially as the “player’s cards”, followed by three more cards as the “flop”. The different sets of cards should be known somehow within your program.

Using the five cards that are drawn in total, the program should determine the strongest poker hand out of the following options *(Ranked weakest to strongest)*:

1. High card
2. One pair

3. Two pair
4. Three of a kind
5. Straight
6. Flush
7. Full house
8. Four of a kind
9. Straight flush
10. Royal Flush

Note: It might be a good idea to understand the requirements for each poker hand before starting the program.

Once this set of five cards has been checked for, add one more card to the set, making the total six cards. With these six cards, you should be able to choose the five best cards that are relevant for each separate poker hand. This means that the group of five cards can vary based on the different hands that are being checked for.

Again, your objective is to determine the strongest poker hand out of the 10 possibilities.

This process will be repeated one more time with one additional card, making the total count seven. Once again, you should choose the five best cards out of the seven that are specific to the current poker hand you are checking for (to see whether or not it exists). Your objective is to determine the strongest poker hand.

Presentations

Teams will create and present a short 2-3 minute presentation and demo for our judges, followed by a 2-3 minute Q&A period. Parts of the presentation should be shared equally between the team members to score full points. Judges may ask questions following the presentation and demo.

In your presentation, you should give a demo of your program, discuss the structure of your solution and any major design decisions that were made during the development.

Marking Scheme

The final team score is based on quantitative evaluation of the product based on the following criteria. Basic software engineering principles, best practices and quality of the team presentation will all be taken into consideration.

Criterion	Poor (<50%)	Good (50% - 75%)	Excellent (75% - 100%)	Score
Software Design				
Code structure	Software structure is haphazard and disorganized. Code is not separated along logical boundaries and may all be contained in a huge monolithic class or function.	Software structure is fairly well-designed. Code is partitioned into smaller functions or classes where appropriate. Attempts are made to design for code modularity.	Software structure shows evidence of planning and design. Architecture follows good design principles and is modular and extensible.	**/10**
Code cleanliness	Code is poorly formatted. Use of whitespace is inconsistent. Variable and function names seem confusing or arbitrary.	Code is mostly consistent and readable. Some areas have poor naming or styling. Comments are lacking for complicated sections that would benefit from explanation.	Code is immaculately formatted throughout and is immediately readable and understandable. Names are chosen appropriately and comments are added when necessary.	**/10**
Completion of product	Limited code is displayed in file, which shows a limited understanding of the program and therefore resulting in an incompletion of the product	Code mostly translates to the completion of the product. Majority of the functions are working with minimal errors.	Program runs smoothly. There are no errors that occur when the program is run multiple times for different sets of cards.	**/5**
Total				**/25**
Presentation				
Explanation of design decisions	Solution is presented poorly. Explanation of solution structure and algorithms are missing or unclear. Little or no justification is provided for design decisions.	Solution is explained adequately. Figures and diagrams are used where appropriate. Some design decisions lack justification.	Solution is explained clearly and eloquently. All design decisions are explained and justified.	**/5**
Demonstration of final product	Demonstration was poorly done. Many components were unclear and did not execute properly.	Many components of the final product were demonstrated well with minimal errors.	All components of the final product were successfully delivered. The final product executed perfectly with no errors.	**/5**
Communication	Presentation is	Presentation is mostly	Presenters are professional	**/5**

	disorganized and unpracticed.	organized with only the occasional stumble or misstep.	organized, and articulate.	
Total				**/15**
Grand Total				**/40**

In case of a tie in total marks, the judging panel will confer and select the winning team. Completed marking sheets will not be disclosed to competitors; however, if teams wish to know their strengths and weaknesses for improvement in future competitions, judges and FIrst Year Directors may be available after the competition for questions.