**ASIA PACIFIC UNIVERSITY TENNIS CHAMPIONSHIP MANAGEMENT SYSTEM**

*(Working Duration: Monday of Week 10 to Monday of Week 14 – 30 Marks)*

**STORY BACKGROUND:**

The **Asia Pacific University Tennis Championship (APUTC)** has become a premier event in the world of collegiate tennis, attracting top players from all over Asia. Hosted annually by Asia Pacific University (APU), this tournament is more than just a test of athletic ability; it's a celebration of the spirit of competition, sportsmanship, and innovation. Over the years, the championship has evolved into a highly efficient, technologically advanced event, making use of cutting-edge data structures to streamline operations.

The leadership at APU, always passionate about both education and technology, recognized that organizing a tournament of this scale required a balance between human ingenuity and sophisticated tools. They envisioned a seamless, fair, and efficient experience for both players and spectators. With this goal in mind, APU’s team decided to implement several advanced data structures - ***Stack****,* ***Queue****,* ***Priority Queue****, and* ***Circular Queue***—to address the unique challenges posed by a tennis championship.

**CHALLENGES:**

Managing a tennis championship involves overseeing several complex system features, including:

* **Tournament Scheduling and Player Progression**, which includes organizing match schedules and tracking player advancement through stages such as qualifiers, round-robin, and knockout rounds.
* **Ticket Sales and Spectator Management** to guarantee fair access to tickets, prioritizing VIPs and early-bird buyers.
* **Player Withdrawals** due to injuries or other reasons.
* **Match History Tracking** to record results and updates as the tournament progresses.

**System Tasks That Can Be Allocated to Each Member for This Assignment**

Managing a tennis championship requires coordinating multiple complex elements to ensure smooth operations. Each of these components involves specific tasks that can be optimized using suitable data structures such as ***Stack, Queue, Priority Queue, and / or Circular Queue***. The following tasks are essential for the success of the tournament:

**TASK 1: TOURNAMENT SCHEDULING AND PLAYER PROGRESSION**

This task involves organizing and coordinating the match schedules and tracking players' advancement through different stages of the tournament. These stages include qualifiers, round-robin, and knockout rounds.

**Key Actions:**

* Schedule matches in a way that allows players to compete in the right order.
* Track player performance to determine progression through tournament stages.
* Ensure the correct matchups are created for each round (e.g., qualifiers to knockout rounds, winners advancing).

**TASK 2: TICKET SALES AND SPECTATOR MANAGEMENT**

This task focuses on ensuring fair access to tickets and managing spectator entry, prioritizing VIPs, early-bird buyers, and public access.

**Key Actions:**

* Process ticket sales efficiently while giving priority to VIPs and early ticket purchasers.
* Manage spectator flow to ensure a smooth entry and exit process at the venue.
* Ensure that ticket sales are well-organized and prevent issues like overcrowding or confusion at ticket counters.

**TASK 3: HANDLING PLAYER WITHDRAWALS**

In a high-profile tournament, player withdrawals due to injuries or other reasons need to be managed quickly to maintain the integrity of the tournament.

**Key Actions:**

* Track player withdrawals in real time and update the tournament bracket accordingly.
* Make necessary adjustments to the match schedule and opponent assignments for players who remain in the tournament.
* Ensure that any replacements (e.g., substitute players) are added in an organized and fair manner.

**TASK 4: MATCH HISTORY TRACKING**

This task involves recording match results, updating scores, and keeping a historical record of the tournament as it progresses.

**Key Actions:**

* Record match results, including player scores and outcomes.
* Update and maintain a historical log of match data for reference and reporting.
* Provide easy access to historical data for analysis and potential disputes resolution.

**Lab Work #2 – Program & Live Presentation Guidelines (30 Marks)**

1. A team can only contain a maximum of **FOUR (4)** members.
2. Your team is required to use C++ programming to develop **ONLY ONE (1)** prototype in this section.
3. **Built-in containers such as <list>, <vector>, etc. are not allowed** in this assignment. All containers are self-created.

*Refer to the link:* [*https://www.geeksforgeeks.org/containers-cpp-stl/*](https://www.geeksforgeeks.org/containers-cpp-stl/) *for further information on built-in containers in STL C++.*

1. Each team member must be responsible for **at least ONE (1)** of the following tasks, applying the appropriate data structures and algorithms.
   * **Task 1: Tournament Scheduling and Player Progression**
   * **Task 2: Ticket Sales and Spectator Management**
   * **Task 3: Handling Player Withdrawals**
   * **Task 4: Match History Tracking**

*Marks awarded will be based on individual contributions, considering each member's responsibility in the system and how accurately you can justify your selection of data structures and algorithms.*

1. The evaluation criteria for this lab work #2 also include assessing the clarity and structural design of the code, as well as the quality of comments and adherence to good programming practices. (e.g., indentation, meaningful identifier names, comments, etc.).
2. **This task requires a group submission, but grading will be based on everyone’s contribution to the system.**

The team leader must upload a ZIP file of the system solution to the Moodle system by **Monday of Week 14**, no later than 5:00 pm.

The zip file must adhere to the following name format:

*“<GroupNo>\_<student ID-leader>\_<student ID-member1>\_<student ID-member2>\_<student ID-member2>.zip”*

For example, **“G1\_TP012345\_TP012344\_TP012123\_TP012111.zip”**

Refer to **Page 6** for marking criteria of this Lab Evaluation Work #2 submission.

1. After submitting your system code to Moodle, your team must schedule ***a live presentation*** with your lecturer *between Tuesday of Week 14 to Friday of Week 16 (Your final exam week #1).*

**Summary: What Do You Need to Hand in During this Assignment Submission?**

1. This assignment requires **TWO (2)** submissions by your team, which include the following:
2. **Lab Work #2 – Group Submission**

* C++ solution in zip folder, inclusive *the .cpp, .hpp and csv/text files*.

1. **Individual Live Demonstration (30 Marks)**

* Each member must present their specific contribution to the system.
* The presentation should be completed within 30 minutes, including both the **system demonstration and a Q&A session**.
* PowerPoint slides are not required for this demonstration.

1. Your team will need to submit all your C++ solutions to the Moodle system by **Monday of Week 14, before 5:00 pm**, and arrange *a live presentation with your lecturer between Tuesday of Week 14 and Friday of Week 16*.
2. **Missing the live presentation** will result in your assignment task #2 receiving **0 marks**.

MARKING CRITERIA

(Lab Evaluation Work #2 - 30 MARKS)

This Lab Evaluation Work #2 will be assessed based on the following INDIVIDUAL performance criteria:

|  |  |  |
| --- | --- | --- |
| **Assessment Components** | **Inclusive** | **30 Marks** |
| *CLO3: Lab Evaluation Work #2 – Individual Development Skills* | | |
| **Practical Skills: Problem-Solving Skills (15 Marks)** |  |  |
| * Identify and address technical challenges. | Assessment of Problem-Solving Ability |  |
| * Use of data structures and algorithms. | Technical Proficiency |  |
| * Implementation of features according to design specifications. | Technical Proficiency |  |
| * Code quality, including readability, efficiency, and correctness. | Code Quality Evaluation |  |
| * Quality of individual contributions relative to team goals. | Contribution Assessment |  |
| * Innovation and creativity in developing and implementing features. | Creativity and Innovation Evaluation |  |
|  | | |
| **Practical Skills: Q&A with Justification of Data Structures (15 Marks)** |  |  |
| * Clear and logical explanation for the choice of data structures. | Justification Ability |  |
| * Relevance of chosen data structures to the functionality implemented. | Relevance Evaluation |  |
| * Justification aligned with the system requirements and performance needs. | Alignment with Requirements |  |
| * Effectiveness of the live presentation in explaining individual contributions | Presentation Effectiveness |  |