Neumann’s Matrix

Module Introduction

Domain Overview

The Computer Science module is designed to simulate how computing skills are applied in fast-paced, real-world problem-solving environments. Across three progressively complex rounds, delegates move from algorithmic speed and precision, to analytical data reasoning, and finally to investigative cybersecurity thinking. The module emphasizes not just coding ability, but adaptability, logical reasoning, and strategic decision-making under pressure.

Journey Summary

Delegates begin with **Round 1: Code Rush**, a high-intensity competitive programming round focused on speed, correctness, and algorithmic thinking. Those who qualify advance to **Round 2: DataForge**, where the emphasis shifts from writing code quickly to understanding data, diagnosing flawed machine learning pipelines, and optimizing model performance. The final stage, **Round 3: CipherQuest**, challenges delegates to think like cybersecurity investigators, solving interconnected puzzles through observation, logic, and light exploitation techniques. Together, the three rounds test a delegate’s breadth across core CS domains: programming, data science, and cybersecurity.

1. Round 1: Code Rush

Delegate Cap(2 delegates)

Overview & Significance

Code Rush is a speedcoding round hosted on **DOMjudge**, a platform commonly used in competitive programming contests. This round evaluates a delegate’s ability to quickly understand algorithmic problems, implement correct solutions, and manage time effectively. It represents the foundational CS skill set: turning problem statements into working logic under strict time constraints.

This round is significant because it filters delegates based on their core problem-solving instincts, coding accuracy, and ability to perform under pressure—skills essential in both academic CS and industry interviews.

Step-by-Step Progression (Delegate Viewpoint)

1. Login & Environment Setup
As a delegate, you receive your team’s login credentials before the round. At the venue, you

are given the DOMjudge server URL. You open the link in a browser, log in, and see the contest dashboard. The contest is visible but locked until the official start time. This is your window to ensure your laptop, IDE, and language setup are ready.

2. Contest Start – Problems Released
At the start signal, all problems unlock simultaneously and the contest timer begins. You immediately see a list of problems, each with a full problem statement, input/output format, and constraints. You skim all problems quickly to assess difficulty and decide which ones to attempt first.

3. Solving & Submitting
You write solutions locally in your chosen programming language. Once ready, you upload the solution file through the DOMjudge submission panel. Within seconds, you receive a verdict such as Accepted, Wrong Answer, Runtime Error, or Time Limit Exceeded. The live scoreboard updates continuously, reflecting your performance relative to other teams.

4. Interactive Moments
During the contest, your team may be selected for short interactive challenges. These are designed to add unpredictability and energy to the round. They may temporarily change how you code or provide minor advantages or constraints. You are expected to adapt quickly without losing focus on problem-solving.

5. End of Contest
When time expires, submissions close automatically and the scoreboard freezes. Your final rank is determined by the number of solved problems and penalties from incorrect submissions.

Evaluation Focus

Delegates are assessed on:

- Algorithmic correctness and efficiency
- Ability to interpret problem statements accurately
- Time management and strategic problem selection
- Clean handling of edge cases

Logistics & Required Tools

Provided Resources

- DOMjudge contest server
- Problem statements and automated judging system

Delegate Requirements

- Personal laptop
- Any local code editor or IDE
- Familiarity with at least one competitive programming language
- Understanding of basic DSA (ie Sorting, Searching, Pathfinding, String Manipulation, Mathematical Problems)

Elimination Rate
Approximately **50% of delegates** are eliminated after this round.

2. Round 2: DataForge

Delegate Cap(3 delegates)

Overview & Significance

DataForge is an applied machine learning and data analysis round conducted entirely in **Jupyter Notebooks**. Instead of building models from scratch, delegates are given intentionally flawed notebooks containing imperfect analysis, preprocessing, and models. The goal is to understand what is wrong, improve it logically, and extract the best possible performance.

This round is significant because it tests real-world data science skills: reasoning about data, debugging analytical workflows, and making informed modeling decisions rather than blindly coding.

Step-by-Step Progression (Delegate Viewpoint)

1. Access & Initialization
You open your assigned Jupyter environment where all required libraries are pre-installed (Done by the delegate before the round using mentioned tutorials). Three files are provided: one notebook for data exploration, one for model training and tuning, and CSV files containing training and test datasets. Before making changes, you ensure everything runs without errors.

2. Orientation Briefing
You receive a short briefing explaining the dataset context, the target variable, the submission format, and the evaluation metric. No hints are given regarding the best model or approach, so understanding the problem definition is critical.

3. Data Exploration Phase
You explore the dataset using visualizations and statistics made by matplotlib, seaborn and pyplot. You identify noisy features, correlations, missing values, and patterns that may affect model performance. Decisions here guide what preprocessing and feature selection changes you make later.

4. Model Tuning Phase
You work with pre-built models already present in the notebook based on python's scikit-learn library. Performance is improved by fixing preprocessing logic,
adjusting
hyperparameters, and comparing results using validation data. Thoughtful experimentation is rewarded more than random trial-and-error.

5. Final Prediction & Submission
Once satisfied, you generate predictions on the evaluation dataset and export them in the required format. You submit the output to the scoring system. Formatting errors can invalidate an otherwise strong solution.

6. Round Conclusion
After submissions close, scores are calculated and ranked based on model performances (MSE + MAE + R2 Normalized for best 3 models)

Evaluation Focus

Delegates are assessed on:

- Quality of data reasoning and feature selection
- Understanding of machine learning workflows
- Logical tuning of models and preprocessing
- Ability to maximize performance within constraints

Logistics & Required Tools

Provided Resources

- Jupyter Notebook environment
- Pre-written analysis and modeling notebooks
- Training and test datasets (CSV)

Delegate Requirements

- Personal laptop
- Comfort navigating and editing Jupyter Notebooks
- Fundamental understanding of ML concepts

Elimination Rate
Approximately **40% of remaining delegates** are eliminated after this round.

3. Round 3: CipherQuest

Delegate Cap(3 delegates)

Overview & Significance

CipherQuest is a cybersecurity Capture-the-Flag (CTF) style round focused on investigation, logic, and lightweight exploitation. Delegates solve a chain of interconnected challenges by uncovering hidden information, decoding clues, and identifying vulnerabilities.

This round is significant because it tests a different CS mindset: curiosity, attention to detail, and the ability to connect small clues into a coherent solution path.

Step-by-Step Progression (Delegate Viewpoint)

1. Initial Investigation
You receive a single starting resource, either a web link or a downloadable package. You carefully inspect it for suspicious files, hidden text, unusual behavior, or vulnerable components. Small observations often unlock the next step.

2. Exploitation & File Analysis
Successful investigation leads you to server-side files, databases, archives, or protected resources. You decide how to proceed based on clues rather than explicit instructions.

3. Multiple Solution Paths
At various points, you choose between brute-force/manual approaches or smarter shortcuts discovered through exploration or OSINT. Both work, but efficiency and speed are crucial.

4. Puzzle Solving & Decoding
You encounter encoded strings, simple ciphers, steganography, image metadata, and logic puzzles. Solving these reveals credentials, passwords, or new locations to investigate.

5. OSINT & External Clues
Some challenges require searching linked pages or public resources created as part of the challenge. Not all information is hidden in code.

6. Final Flag Discovery
Correctly combining clues leads to one or more flags, which you submit through the scoring system. Rankings are based on correctness and speed.

Evaluation Focus

Delegates are assessed on:

- Observational and investigative skill
- Logical reasoning and pattern recognition
- Efficiency in choosing solution paths
- Ability to connect disparate clues

Logistics & Required Tools

Provided Resources

- Starting link or challenge package
- Flag submission system

Delegate Requirements

- Personal laptop
- Web browser and basic file analysis tools
- Willingness to experiment and explore

Elimination Rate
This is the **final round**, determining overall rankings and winners.

4. Preparation & Study Resource Hub

Round 1:

1. Python Tutorial
2. C++ Tutorial

Round 2:
Overview:
GeeksForGeeks

Scikit Learn Documentation:
How to set up Jupyter Notebooks + Python in VSCode
1. Polynomial Regression (via PolynomialFeatures)
2. Random Forest Regression (RandomForestRegressor)
3. MLP Regression (MLPRegressor)
4. Elastic Net (ElasticNet)
5. Support Vector Regression (SVR)

Round 3:

1. Dcode
2. ExifTool
3. Base Conversion + Hashing
4. Steganography
5. CyberChef