

Data Science C

1. **DESCRIPTION:** Teams will demonstrate a theoretical and practical knowledge of programming, with a focus on data analysis techniques.

| | |
|---------------------------------|--------------------------------------|
| <u>TEAMS OF UP TO:</u> 2 | <u>EYE PROTECTION:</u> None |
| <u>IMPOUND:</u> None | <u>EVENT TIME:</u> 50 minutes |
2. **EVENT PARAMETERS:** Teams will complete both a written test and an on-site series of Coding Challenges in Python 3.7 using the Jupyter Notebook interface.

 - a. Teams must bring writing utensils. Teams may bring a single calculator (which must not be capable of internet access). Teams may bring any amount of printed or written notes.
 - b. The event supervisor will arrange to provide each team with a single computer which is capable of editing and running Jupyter Notebooks with Python 3.7. Any necessary external packages (e.g. NumPy, scikit-learn) will be installed.
 - c. Internet access is forbidden at all times. Violation of this rule will result in immediate disqualification.
3. **THE COMPETITION:**

 - a. **Written Test.** Teams should demonstrate a theoretical understanding of the following topics.
 - i. **Python 3 syntax and programming knowledge.** Includes but not limited to: variables, data types, control flow, methods/functions, classes, object-oriented programming, basic data structures, time complexity analysis.
 - ii. **Statistical methods:** common statistical distributions, central tendency and dispersion, correlation and covariance, joint/marginal/conditional distributions, visualizing data, central limit theorem, confidence intervals/z-score.
 - iii. **Feature extraction and learning:** high-dimensional data, principal component analysis, k-means, binary classification (decision trees, k-nearest-neighbors, SVM), linear regression, gradient descent, neural networks.
 - b. **Coding Challenges.** Each Coding Challenge is a Jupyter Notebook with some starting code.
 - i. A Coding Challenge is a programming problem, and a solution is a completed Jupyter Notebook which processes some data and outputs the appropriate result. The notebook will contain detailed descriptions of the problem, the input data format, and the expected output format.
 - ii. All input data and expected outputs will be provided to teams. Teams' solutions should be evaluated only on the inputs provided to them.
 - iii. The supervisor will provide a partially completed solution. In particular, the provided notebook should include code that provides input and automatically evaluates the correctness of the output. The provided notebook may also

outline the intended structure of the solution (if there are multiple functions/classes involved, etc.).

- iv. Teams should submit their solutions according to supervisor instructions.
- v. If teams are expected to use external libraries, the supervisor will provide the API. Useful external libraries are math, NumPy, and scikit-learn.

4. **SCORING:** High score wins. Written test: 60%, Coding Challenges: 40%.

- a. Coding Challenges will be scored based on correctness (75%) and readability (25%). Intentionally obfuscated code gets 0 points. Any solution that hardcodes the outputs gets 0 points.
- b. Readability entails following major Python conventions, using well-named variables, and organizing code cleanly. The supervisor should be able to easily follow the logic of the code. Teams may add code comments as they see fit.
- c. Tiebreakers (in order of priority): highest Coding Challenges subscore, having well-documented code, first written test question missed.

5. **RECOMMENDED RESOURCES:**

- a. <https://codingbat.com/python>
- b. <https://leetcode.com/problemset/all/>
- c. <https://www.codecademy.com/learn/learn-statistics-with-python>
- d. <https://towardsdatascience.com/beginners-guide-to-machine-learning-with-python-b9ff35bc9c51>
- e. https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi