**Python Code Analysis**
Trial Event Rules 2018

1. **DESCRIPTION**: Given a set of computer programs or parts of programs, teams will determine either the output or the error that each would produce.
   a. **A TEAM OF UP TO**: 2 **APPROX TIME**: 50 minutes

2. **EVENT PARAMETERS**:
   a. Each team is permitted either one 8.5×11 inch, double-sided reference sheet or two 8.5×11 inch single-sided reference sheets.
   b. Competitors may NOT bring electronic devices into the testing room, including computers, tablets, cell phones, smart watches, or calculators.
   c. All computer code shall be given in the Python programming language (version 3.x). Problems should focus on general programming principles rather than the peculiarities of the Python language.
   d. Code may be given as a sequence of statements to be executed in order, as a method or function, or methods/functions to be called with a given set of arguments.
   e. All input and data values shall be specified in the given code or as a given set of arguments, external files, or graphical user interfaces.
   f. Each problem will take one of the following two forms. Competitors shall:
      i. Determine the output of the code, either the return value of a method/function call or the text created by a print statement. Space shall be provided for competitors to write out their answer. If a problem's score depends on the number of output lines or the number or spacings of characters, then the answer space shall include a grid.
      ii. Describe the run-time error that results when the given code runs. Competitors are not expected to reproduce the error message or exception type, but rather to describe concisely and specifically what goes wrong, why the code does not run properly, and how to correct it.

3. **PERMITTED TOPICS**:
   a. Integers, unary operator -, and binary operators +, -, *, /, and %. No more than 10% of the score may involve bitwise operators ~, &, |, ^, <<, or >> , or binary, octal or hexadecimal constants.
   b. Characters must be in the range ' ' (space) to '~' (tilde), plus escape sequences '\\' (backslash), '\'' (single quote), and '\"' (double quote).
   c. Boolean True, False, unary operators 'not' and type casts, binary operators ==, !=, <, <=, >, >=, 'and', 'or'; inline if statements; and operator precedence and associativity.
   d. Lists, initializing lists, setting and getting elements using [], modifying lists using insert and append, list comprehension, and sorting lists.
   e. Strings: string literals made from the permitted character literals and methods len, endswith, startswith, find, index, count, join, split, replace.
   f. Assignments: = or +=, -=, *=, /=, &=, |=.
   g. Conditionals: if, if … else, while, for. Any style of comments. Code shall be properly indented, except in problems amounting to no more than 10% of the score.
   h. Print formats is limited to fixed text and format specifiers %%, %s, %c, %d, %f, %.<number of digits>f, \n.

4. **FORBIDDEN TOPICS**:
   a. Import statements and class definitions may be included to make the code a complete program but competitors are not expected to interpret their significance.
   b. Classes, interfaces, inheritance, polymorphism, constructors, lambdas, method overloading, regex/regular expressions.
   c. File I/O, networking, threads, web programming, dynamic memory management.
   d. Any feature of Python that is non-standard or which varies among versions or platforms.

5. **SCORING**:
   a. Each problem shall specify its number of points. Partial credit should be given where appropriate.
   b. No single problem may be worth more than 20% of the total score.
   c. Some questions will be designated as tie breakers.