

WHAT THE FUNCTION

1. **DESCRIPTION:** Teams will, given code fragments in the Python 3 programming language, answer questions about the code, using concepts from both math and computer science. There is an emphasis on **functions and lambda expressions**.

A TEAM OF UP TO: 2

EVENT TIME: 50 minutes

2. **EVENT PARAMETERS:**

- a. Each team may bring one 8.5" x 11" sheet of paper, with any desired information on both sides of the paper.
- b. Scratch paper will be provided by the event supervisor.
- c. The test will be conducted on paper, without the use of computers. Competitors may not bring any electronic devices into the testing room, including phones, computers, calculators, and smart watches.

3. **THE COMPETITION:**

- a. Teams will not be required to write code, aside from small fragments (up to a line long)
- b. Questions will require knowledge of basic Python 3 syntax, including to the following:
 - i. Variable assignment, including assignment with operators such as +=
 - ii. Functions, **including lambda expressions and using functions as variables or arguments of other functions**
 - iii. Control structures: **if, elif, else, for, for/in, while**
 - iv. Types **int, str, list, dict, bool**, including basic operations such as indexing, substring/sublist, insertion
 - v. Unary operator **-**, binary operators **+, -, *, /, //, **, %, ==, !=** (all but the last two are on integers only except **+** and *****, which may be used with strings and lists)
 - vi. Boolean operators **and, or, not**
 - vii. Built-in functions **print, chr, ord, sum, len, max, min, pow, abs, range, map, filter**
- c. Questions may also test basic computer science concepts such as runtime analysis and big O notation.
- d. The following topics will **not** be tested:
 - i. Control structures: **try/except, continue, break, pass**
 - ii. Operators used with non-standard types (e.g., **1 + True**)
 - iii. Types and built-in functions not previously listed (including **float**)
 - iv. Standard modules
- e. Some types of questions may include:
 - i. Given a function, find the input that will produce the desired output.
 - ii. Find the input that makes a function perform a maximal number of steps

4. **SCORING:** Point values will be indicated on each problem. Problems to be used as tiebreakers will be selected beforehand.