

Source Code

2017 Trial Event B/C – Michigan Science Olympiad

1. **DESCRIPTION:** This event combines computer science and mathematics to solve problems. The objective is to develop and code an efficient computer program that finds solutions to these problems.

A TEAM OF UP TO: 2

IMPOUND: None

EYE PROTECTION: None

APPROXIMATE TIME: 50 Minutes

2. **EVENT PARAMETERS:**

- A. Each team will be provided one computer. Computers will have no Internet access.
- B. Teams may not use any other electronic devices besides the provided computer.
- C. Teams will be allowed to bring one 8.5" x 11" sheet of paper (both sides) of notes.
- D. The Programming Language used will be Python. (Version 3.4.3 or newer)
- E. A Python IDE or editor will be installed on the computers. This should be announced before the event.
- F. Student's completed Python files will be submitted. Details will be provided during the event.

3. **THE COMPETITION:**

- A. Students should know basic syntax of the Python programming language and mathematical concepts described in the categories below (Section D and E). Event problems will test these skills.
- B. Event supervisors will provide teams with a python file containing a main function and function definitions for each problem. The main function will call each problem's function and print the returned value. The problems will be described in comments above their corresponding function. Teams will write their code for each problem in the related function using their logic to solve the problem. Functions should be structured such that nothing is output using a print statement or otherwise. The answer to the problem should be returned using a return statement. The main function should not be edited by the teams.
- C. In the spirit of the event, all answers should be derived programmatically (unless the problem is multiple choice). While not required, commenting code is a necessity for good organization.
- D. Python Programming Language Topics
 - a. Types and Functions (numbers, strings, type conversion, slicing, len(), etc.)
 - b. Arithmetic/Boolean Operations (and, or, not, equal, addition, multiplication, modulus, etc.)
 - c. Control Flow/Logic (if, elif, else, while, for, switch, etc.)
 - d. Data Structures (Arrays, Dictionaries, Tuples)
 - e. Import statements / Libraries.
- E. Mathematical Topics
 - a. Order of Operations, Powers, Factorials, Averages
 - b. Area and Volume of shapes (Basic Geometry Concepts)
 - c. Prime Numbers, Fibonacci Numbers (Basic Algebra Concepts)

4. **SCORING:**

- A. Highest score wins
- B. Points are earned as follows:
 - a. Each problem will be worth a designated number of points. Partial credit may be given.
 - b. Programs that do not run/partially run (ie syntax errors or otherwise) will only be able to receive points until the point of failure. Teams should validate and run their code frequently.
 - c. Use of other electronic devices (besides the provided computer) will result in disqualification.
- C. Tie Breakers
 - a. First tie breaker will be the team correctly solving a designated tie breaker problem.
 - b. Second tie breaker will be the team with the highest number of correct problems.
 - c. Third tie breaker will be the team with the fewest lines of code in the most difficult problem answered correctly. (And subsequent problems if ties persist) Commented lines will not count.

ADDITIONAL RESOURCES: <https://www.python.org/> | <https://www.codecademy.com/>