Source Code - How To Run / Supervise

Trial Event – Michigan Science Olympiad

1. **The Supervisor**
    a. The supervisor should be someone with programming and software development experience. The ability to solve problems is a good trait.
    b. They should have an understanding of the Python programming language.

2. **Event Resources & Materials**
    a. Each team should be provided with a computer.
        i. Computers should have no Internet access.
        ii. Computers should have a Python IDE or editor installed. Python installations come default with the IDLE IDE. Teams may want to know in advance exactly which software will be installed. A simple IDE is preferable. Code completion and logic assistance functionality is discouraged.
        iii. If an issue arises with a computer during competition, a supervisor should be able to solve the problem or provide the students with new computer.

3. **Designing the Event**
    a. The Python program is the event. As described in the rules, each function represents one problem with the addition of a main function (which calls all the other functions).
    b. All problems should require the teams to return a single value. Problem instructions are to be written in a comments section above each function.
    c. Problem Types
        i. Multiple choice problems. The choices would be defined in comments and the teams would return the correct choice's corresponding letter. Examples:
            1. Python Syntax Questions
            2. Logic Questions
            3. Correcting incomplete or incorrect code fragments
        ii. Code Implementation problems. These problems will be more time consuming, so it is important to not make the solution too lengthy. Students will implement their code/algorithm and programmatically return the solution to the problem. Examples:
            1. Solving math questions impossible to be solved quickly be hand.
            2. Using logic to derive the correct solution.
    d. Assigning points to problems.
        i. Each problem should be worth at least two points
        ii. It is recommended that multiple choice problems are worth 2 points while programming implementation problems are worth more based on their relative difficulty.
        iii. For programming implementation problems, bonus objectives can be defined. For example, if the students use a 'for' loop or some other programming concept/element and their returned answer is correct, they will earn the points for the problem plus the bonus points.

iv. It is recommended that there is at least 80 points possible. The more points the better. About half the points should come from multiple choice questions and the other half from programming implementation questions.

v. Tie breakers are defined in the rules.

e. View the sample Source Code program for examples.

4. **Moderation of the Event**

a. The Entirety of the event and student's work will take place in the Python program on the provided computers. The supervisor should have a method of starting and stopping the event at the same time for all teams. Examples:

i. Locking/Unlocking the computer.

ii. Verbal instructions on starting/stopping (where to access the program file).

b. Giving a five to ten minute warning before the end of the event will give the students a chance to run and validate their code before submission.

c. Supervisors can assist students with computer issues, but should not help the students solve the problems for the event.

d. Any team using resources other than their computer or their index card of notes should be disqualified.

i. Cell Phones, Tablets, calculators, etc are not allowed.

e. Where possible, computers should be positioned in a way where teams have no visibility to other teams displays. A clear demonstration of cheating will also result in disqualification.

f. Event Submission. Where to submit each teams program file. (Supervisors choice)

i. Shared Network drive

ii. Collection on USB drives

5. **Scoring the event**

a. The supervisor can manually run each teams program and compare the output with their correct program to score. The supervisor may also find it useful to implement their own program to automatically score each teams program.

b. See the Source Code rules for a complete guide to scoring.

6. Extra Notes

a. Teams are given about 50 minutes to compete in the event. Having more than 50 minutes worth of problems / content is encouraged. This way supervisors can maximize the total number of points and create a larger variance between team's raw scores.

b. Keep the majority of the problems simple and direct.