



---

## 2009 HIGH SCHOOL PROGRAMMING CONTEST RULES

### **Programming Languages**

This year's contest will support the following programming languages and environments on PC's running Windows Vista: QBasic, Visual Basic .NET, Visual C/C++ .NET, Visual C# .NET, Sun Java 2 JDK 6 with Eclipse IDE. Teams may program in any or all of these languages. The request for language information on the registration form is to help us assign teams to labs.

With the exception of Visual Basic and Visual C#, all programs are expected to be written as console applications with all code residing in a single file. Visual Basic and Visual C# programs also may be console programs, but we are not requiring it at this time. Visual Basic and Visual C# programs using a GUI for input will be expected to follow the form of the console applications as much as possible. This includes labels for the prompts, input text boxes for user input, a Submit button to cause the required calculations to be performed, and a labeled text area showing the results. GUIs not sufficiently labeled or particularly difficult to use will be judged to be incorrect (see below).

Although Java is not intended for writing console programs, the problems that we pose for the contest are intended to be solved this way. We are interested mostly in the problem-solving aspects of the contest problems and not as interested in the user interface aspects. For example, we do not expect exact adherence to formatting, though we do expect something close. It is relatively simple to do console input using the Scanner class with System.in or the BufferedReader class with System.in and the parsing routines of the Integer and Double classes for input. Console output can be performed using System.out.println. All of these routines are standard Java. See the sample Java programs posted at the contest website for details.

### **Submission Guidelines**

Submissions are made electronically via a submission Web page. Detailed submission procedures will be made available and demonstrated during the initiation and practice session.

A submission is judged by running it on some test data (in addition to the sample data given in the problem). If the submission gives the correct answers for all test cases, the submission is judged solved and the team will be so informed. If the submission does not give the correct answer for all test cases, the submission is judged incorrect and an error message is returned to the team. The error messages and their meaning are as follows:

Compilation error	The program failed to execute due to syntax errors
Run-time error	The program crashed or entered an infinite loop, possibly producing one or more correct or incorrect answers before doing so
Wrong answer	The program produced an incorrect answer for one or more of the judges' test cases
Presentation error	The program fails to provide sufficient input prompts or output labeling, or requires awkward input. No effort will be made to determine if the program produces correct or incorrect answers.

Generally, we do not expect there to be any compilation errors or presentation errors. However, compilation errors sometimes happen when a file is submitted before it is saved. And a particular example of a presentation error from the recent past was a GUI Visual Basic program that required each character of a 2-D array of characters to be entered in a separate input text box.

**Multiple submissions may be made for each problem until the problem is judged solved or the contest ends, whichever comes first.**

### Scoring

**The team solving the most problems is the winner.** In case of ties, a tie-breaking point system will be used. The point system is based on elapsed time and number of submissions. Lower point totals are better (when comparing teams solving the same number of problems).

Each problem submission adds 20 points to a team's score (regardless of whether the solution is judged correct or incorrect). In addition, one point for each elapsed minute is added to the score when a correct solution is submitted.

**Example:** Team A completes Problem 1 30 minutes into the contest. They then complete Problem 2 90 minutes into the contest. On Problem 3, they have one incorrect submission, but submit correctly 120 minutes into the contest. Team A fails to solve any other problems, but does submit an incorrect solution to Problem 4. Team A's score is computed as follows:

Problem 1:	20	1 submission	
	+30	30 elapsed minutes	
	<b>50</b>	Problem 1 total points	
Problem 2:	20	1 submission	
	+90	90 elapsed minutes	
	<b>110</b>	Problem 2 total points	
Problem 3:	40	2 submissions	
	+120	120 elapsed minutes	
	<b>160</b>	Problem 3 total points	
Problem 4:	<b>20</b>	1 submission and Problem 4 total points	

Contest total points = 50 + 110 + 160 + 20 = **340**

## **Other rules**

Each team will have one computer assigned to them in a lab. *Depending on the size of the contest, the organizers reserve the right to limit the number of team members that may be in the lab at the same time in which case all teams will be assigned a separate work space.*

All six problems will be handed out at the beginning of the contest. Teams are allowed to make written inquiries about problems via an oracle Web page that will be made available on multiple computers in the lab. Inquiries do not add any points. Answered inquiries will be posted publicly to the submission Web page, so that all teams may benefit. Detailed inquiry submission procedures will be made available during the initiation and practice session.

**Teams may bring up to two computer or language reference books.** Electronic media is not allowed. Program listings are not allowed (with the exception of those in the books). On-line help provided by the IDEs and the Sun Java documentation will be available. Teams also may bring non-programmable calculators. Scratch paper will be provided, but teams will need to bring pens and/or pencils.

Any team finishing all the problems must leave the programming contest area.

Any conduct deemed unsportsman-like, improper, or disruptive by any judge or room monitor will be referred to the contest coordinator, and the involved teams may be disqualified from the contest. *In particular, teams found playing computer games and/or surfing the Web will be asked to leave the contest.*

Teams may not converse with anyone outside of their own team and the programming contest judges, room monitors, and officials. Teams may not converse with their coach.

A computer and projector in each lab will display the current standings. In addition, the submission Web page contains a graphic with the time remaining in the contest. When solutions are submitted, the confirmation Web page will respond with the elapsed time, giving the point potential of the submission if it judged to have solved the problem.

The scoreboard will be turned off during the last 30 minutes of the contest to provide some suspense during the awards ceremony.

## **Sample Problems**

Problems from previous contests may be viewed at our Website:

<http://csserver.evansville.edu/~acm/progcont/>