Computer Science
Programming Contest

By
Charlotte Scroggs
Former Coach and UIL CS Co-Director

Team Member Requirements

- A programming team must have exactly three members
- If a computer science team has four members, the coach decides which three members participate in programming

Hardware Requirements

- Each team has one computer consisting of a single monitor, keyboard, and CPU
- Teams may use one printer, but a printer is not required
- Flash drives are the preferred media to send solutions to the judges, but some contest sites may use a networked system

Software Requirements

- The computer may be loaded with normal software (such as word processors, etc.)
- Java SDK (6.0 or higher recommended)
- a development environment (such as JCreator, Eclipse, NetBeans)

Permissible Items

- on the computer - the documentation for the Java API's, and the API's themselves
- two PUBLISHED printed textbooks or reference books that are reasonably free of handwritten notes.

Prohibited Items

- Any code written by the contestants prior to the contest such as solutions to practice problems, solutions to prior contest problems, or programs written in class are not allowed
- A book on a CD is not an acceptable reference and should be removed from the computer system.

Contest Basics

- The Programming contest will consist of 12 programming problems of varying levels of difficulty.
- Teams have two hours to complete as many solutions as they can.
- Teams determine the order of submission of their solutions.

CS Meet Personnel

Meet Director Duties

- Appoint a contest director (a coach is good)
- This individual is in charge of the contest and has final say on issues involving judging solutions
- Check conflict matrix for a 2 hour slot for the programming contest.
- Assign a room for the contest – Requirements:
- Minimum of 5 feet of table space, 3 chairs, and a power source for each team.
- A clock visible for all contestants
- Assign a room for the Judging area
- Separate from, but near, the contestants' area to allow a timely flow of solutions and results
- Determine where the contest results will be posted
- Meet with the CS Contest Director (next slides)

Meeting between the Meet Director and the CS Contest Director

- Meeting should be early (by mid-January)
- Determine who will provide computers
  - host school provide computers for each team or
  - each team to provide their own system
- Appoint 1 or 2 additional judges (coaches) who will also act as Assistants to the Contest Director
- Determine what computer system will be used for the judging room (1 or 2 systems for 8 teams)
- Who (and how) will install Java, other contest/judging software if applicable
- Determine what judging platform you will use
  - Compile and run the program and manually check the results against the Judges packet
  - Use the program provided by the UIL
  - Use PC^2

Meet Director Duties

- On the contest date, have the following ready for the contest director:
  - All materials provided by the UIL
  - Runners assigned (1 or 2 non-competitors)
  - Signs
    - for each team with team number and school name
    - Judging, Archive and Scorekeeper stations
  - Scratch paper and pencils
  - Timer or stopwatch

CS Contest Director Duties

- **Appoint an archivist and a scorekeeper**
  - These may be separate or combined positions
  - The Contest Director and/or the judges may do this
  - The archivist backs up solutions submitted by teams PRIOR to being judged
  - The scorekeeper records results of contestant's solutions using the provided scoring program
- **Train the runners prior to the contest start**
  - The runners pick up solutions from the contestants and take them to the judging area to be archived, judged, and scored
  - The runners also return judged submissions to the teams
- **Appoint 1 or 2 room monitors**
  - check teams in
  - seat them
  - Both before and during the contest, circulate in the room to see that all teams are adhering to contest rules
  - Coaches or willing teachers are good room monitors

UIL Provided Materials

- **Documents used to create precontest team packets for each team:**
  - A team number, 1 to n.
  - "Precontest Materials" envelope containing:
    - a copy of the dry run problem (dry run is problem number 0)
    - 13 Run Sheets filled in with the team number and problem number
    - 13 Run Envelopes.
    - 3-5 Clarification Forms
    - Team Verification Form
- **Sealed Packet of Problem Sets (1 copy for each team; use the remainder for judging)**
- **Judging packet:**
  - Sealed envelope containing:
    - Printouts of judge data
  - Judging CD, including
    - UIL judging software
    - Judge data
    - Contestant sample data (make one disk copy for each team)
    - Sample solutions
- **Post-contest team packets for each team:**
  - A disk marked “Post-contest Materials” labeled as “Post-contest Materials – DO NOT OPEN UNTIL AFTER THE CONTEST” containing:
    - the problem set
    - the sample data files
    - the judges data files
  - **Note:** Distribute on UIL Embargo Date

CS Contest Director Duties

- **Prior to the contest day**
    - See that the judging software is installed on the judging stations (Java JDK 7.0 required)
    - Install the scoring program for the Scorekeeper
    - Install the archiving software for the Archivist
    - run the solution to the Dry Run to test the judging software, the Scorekeeper's software, and the Archivist's software
- **Day of the contest:**
    - Put up signs identifying each station and team area
    - Ensure judging, archivist and scorekeeper stations are set up
    - Ensure Precontest Material envelopes and Official Contest Materials envelopes are ready for distribution
    - Train judges, archivist, and scorekeeper on the software and their duties
    - Train runners on their duties
- **As teams arrive or 60 minutes prior to the start of contest:**
    - You or an assistant will check in teams and confirm the members' names
    - give teams their Precontest Materials
    - explain to each team the process for submitting their dry run solutions
    - allow teams to set up their computer system
    - allow teams to complete the dry run and submit their solutions
- **The Dry Run**
    - The dry run is a simple problem to:
        - insure teams and officials understand the contest operations.
        - make sure all aspects of the contest are working correctly.
    - Teams submit the source code for the dry run
        - Runners collect it
        - Archivist archive it
        - Judges judge it
        - Scorekeeper record the results
        - Runners return the results to the teams
- **30 minutes prior to the start of contest**
    - Check for the following
        - All previously written programs removed from the contestants' computer system
        - Teams have a single computer system
        - Teams have no more than 2 approved references
        - Extra equipment, resources, backpacks, boxes, etc. are removed from the contest area (moved to front of room)
- **15 minutes prior to the start of contest**
    - Assemble contestants and coaches for instructions. Go over contest rules and procedures and answer any questions
    - Ask the coaches that are not helping to run the contest to leave
    - Distribute Official Contest Materials

CS Contest Director Duties

- **Conducting the Contest**
    - Announce Start. Contestants may open their contest envelope and have two hours to submit solutions
    - When a team solves a problem, they place the source code on a disk, place the disk in a run envelope with a run slip and give it to runner
    - Runner takes envelope to archivist to archive
    - Archivist gives envelope to judge
    - Judge runs the source code and completes the Run Sheet and gives the envelope to the scorekeeper
    - The scorekeeper records whether it was correct or not, awards the appropriate points, and returns the envelope to the runner
    - Runner returns envelope to the team
- **Contest Procedures**
    - The judging data will include test cases not shown on the problem statement sheet.
        - If the solution is correct, it is accepted and recorded as being correct on the run sheet
        - If it fails, a reason for rejection is marked on the run sheet
    - In either case, the solution disk and the run sheet are returned to contestants
- **Ending the Contest**
    - At the two hour mark teams shall be told to stop and no more submissions are accepted
    - A team may turn in a solution at the stop signal if it is in a run envelope and in the air
    - Complete judging
    - Complete scoring
    - Collect team Verification forms, compare with your scoring, and resolve any discrepancies
    - Collect all official contest materials. These contest materials and the Post-contest team packets may be returned to teams when other official contest materials are returned
    - Announce results

CS Contest Director

- **The contest directors are designated UIL officials, and as such have the authority to make decisions regarding the management of the contest to ensure that it is conducted according to UIL rules.**
- **The Contest Director also has the authority to settle any disputes that might arise in accordance with UIL rules**
- **If a situation should come up that you are not sure how to resolve, or if you have any questions regarding rules, procedures, etc., please feel free to contact Shyamal Mitra at mitra@cs.utexas.edu or (512) 471-9708 if a more immediate response is required**

Coaching Tips

By
Charlotte Scroggs
Former Coach and UIL CS Co-Director

Resources

- UIL Site

http://www.uiltexas.org/academics/computer-science

  - **Rules and General Information**
  - Introduction to the Computer Science Contest (includes general guidelines for programming)
  - Scoring Procedures
  - UIL CS Current Topics List
  - Useful Links
  - IDEs
  - References
  - Programming Practice Problems
  - Third Party Materials
  - Other Contests

Programming Problems - Types of problems

- No input required
- Manipulate Strings and chars
- Process data using loops
- Branching (if, switch statements)
- Use of arrays / ArrayLists
- Problems requiring sorting
- Use of matrices
- Use of Stacks, Queues, Maps
- Use of recursion

Programming Problems - Samples

No Input

- One or two problems requiring no input
- Can generally be solved println statements
- Output must be EXACT
    - Sample 1 UIL
    - Sample 2 Lucky Number

Scanner Class and Output

- Read int from a data file
- Read lines from a data file
- Split a line into data elements
- Format output using printf
    - Sample 3 House Numbers

Strings and chars

- Read char's in a String
    - Sample 3 House Numbers
- Read substrings
    - Sample 4 IP
- Patterns (matches)
    - Sample 5 Duplicate vowels

Ways to Sort

- Arrays.sort
- Collections.sort
- Trees
    - Sample 6 Butterflies

Sample 7. `printf` Basics

- Printing String, int, double variables
- Rounding and printing doubles
- Left and right justification
- Printing in columns

Reading Data

- Reading an unknown number of lines
    - Sample 8
- Reading in a matrix of ints
    - Sample 9
- Printing a matrix in columns
    - Sample 9

Sample 9. Using `TreeSet`

- Enter chars into a TreeSet to eliminate duplicate chars.
- Enter words into a TreeSet to sort.

Practice, Practice, Practice

- Practice contests – on line and local