
2019 U.S. Information Technology Collegiate Conference Application Development Competition

Do not put your name(s) or your school's name on anything that you submit, including comments, file metadata, videos, and images.

Doing so will result in disqualification of your team.

The only identifying information you should use is your team number.

Grand Kids Festival Free Fun!

Visit the following site: <https://www.grandkidsfestival.com/>

Your team has been tasked with developing a contest raffle sign up application. Admission to the Festival, including inflatables, pony rides and a variety of additional activities are FREE to the public. However, sponsors have agreed to supply door prizes to the families that attend. Your application must allow a family to register for prizes, including the selection of prizes that they are interested in receiving. Laptops will be available at the festival for registration. Each family will get one draw for the contest from the initial registration. In addition, families can earn additional draws (draw codes) from participating in events. Each event will provide business size cards with draws codes printed on them so that the family can enter into the system for additional draws. Laptops will be available at the exits to allow families to enter the additional codes for more chances to win prizes.

All storage of information will be placed into text files, not databases. This is to remove any connection problems you might otherwise experience. You must include the text files that your application requires with your complete code. Applications which store and retrieve XML or JSON data will earn more points than applications which store information as comma (or tab) delimited.

The registration form should require a parent to enter the following information: First Name, Middle Initial, Last Name, Userid, Password (encrypted or hashed), address, city, state, zip code, phone number and e-mail address. All fields should verify the proper format of information and that information has been entered in all fields. The verified information must be stored into a text file. In addition the system must keep track of how many draws (chances) the family has earned towards the prizes.

The system must allow a registered family to sign back into the system and enter any draw code(s) they have earned. Remember, they might have earned more than one. The valid codes must also be saved in a file so the administrator can add, change, or delete codes when needed. The system does not need to keep track of what codes a family has entered, however,

once a valid code has been entered, the number of draws for the family must be increased. The number of valid codes should be displayed to the parent and saved in the XML/JSON file.

You will also need to create an administration screen which will allow the administrator to request the random draw of a winner for each prize. Remember that the family picks the prizes that they would like to win. Thus, the administration screen should not only pick a winning family but also make sure that family wants the prize. In addition, remember that one family might have multiple chances (i.e., has entered more than one draw code), while another family might have only once chance. You will need design a way to provide more than one chance per family. (*Suggestion*: Load all families in an array, and repeat loading the family into the array the number of draws they have earned.) This screen must require an administrator to log on with a userid and password that exists within an administrators file. **The administrator userid and password MUST be provided in the assumptions file discussed later.**

Requirements:

1. Object-Oriented Programming approaches are preferred and will receive a bonus above non-object-oriented submissions.
2. All variables, functions, classes and objects must use meaningful names. The program name and file names must also be meaningful.
3. All data must be saved to text files only. Data stored and retrieved as XML or JSON format will receive more points than regular text data. Data files must be stored and accessed from their own folder (Data).
 - a Do not use an absolute file path (i.e., do not specify a path like c:\data\data.txt) in your program code. Rather, use a relative path (such as data\data.txt).
 - b Please create a constant at the top of your code which is used to access the file(s). This will allow the judges to easily change the location, if needed.
4. Exception handling must be included. Throw exceptions for missing files and possible user entry errors (such as invalid draw codes).
5. The userid must be used to determine if a family screen or administration screen displays after signing in.
6. Strong security must be provided. Examples include: Limited use of textboxes (use drop downs, lists, radio buttons, check boxes when possible). When using textboxes, check for security intrusion (such as the user trying the enter SQL or HTML information).
7. Program Documentation – Must provide internal (comments) document for the application, functions, and classes to include: IPO (Input-Process-Output) information.
8. Screen Shots: There is no required number of screen shots. However, screen shots must show ALL working parts of the application. For example (but does not include all requirements) you should have shots demonstrating creating a userid, signing in,

selecting prizes to try to win, generating winners, and signing off. If a screen shot is not included for a part of the application, it is assumed that part of the application does not work. **All screen shots must be placed in one file, such as a Word or PDF document.**

Reminder: Applications without screen shots will not be judged.

9. Efficient/Professional: Efficient code will be given more points than other code that accomplishes the requirements, but is not efficient. Code must be professional, well organized and designed.
10. Assumptions: As is the case when gathering requirements from users, many program details are not included in the provided description. There are cases where you must make reasonable assumptions about how the application should function. Document these assumptions in a Word, PDF, or text file and include them with your solution.
11. **Note for Java submissions:** If you are using Java, in addition to installing Java 11, you may need to install Java11FX (now a separate install) from the following site: <https://openjfx.io/>. The judges will be compiling the Java submissions using Java11 with Java11FX. Any additional tools you use (such as form builders) must use only the libraries contained in these installs. If the program does not compile, it will not be judged.

Tie Breaker (work on if you have time):

1. User documentation.

What to turn in:

Create separate subfolders (under a folder with your team name – i.e., APP_##) for each of the following. Application folder which includes all code including source code and compiled (executable) code. Data folder which includes all XML or JSON data files (with example data). Image folder which include any images used in the application. Documentation folder which includes screen shots (all in one file) and an assumptions file. Then compress all items into one ZIP file (do not use any other compression technique). Non-zip files will NOT be judged!

You MUST submit screen shots of your working code. Failure to provide screen shots will cause your application to not be judged. Even if you do not complete all requirements, you still need to include screen shots of what does work.

Code that produces errors when executed will not be judged.

You should comment out code that 'almost works'.