

Name: _____

Date: _____



ACTIVITY 1

Introduction to Celebrity

In this activity you'll be introduced to and play the game of Celebrity. Then you'll brainstorm different design options for creating a computer version of the game, **including the Game class which contains a play method.**

- Your teacher will divide the class into groups.
- On the given sheets of paper, write down the names of five celebrities.
- Fold each paper and put the folded papers into your group's container along with the papers of all group members.
- Within your group, break into teams of two people, so a group of six students will consist of three teams of two students each. These are the teams you'll be competing against.
- Play a round of Celebrity: On each team one person is the designated reader and the other(s) the guesser. Team 1 will go first, they have one minute. The reader picks a paper at random and can say anything but the celebrity's name to get the guesser to say the name of the celebrity. If the guesser gets the name right a point is earned, and another paper is selected. If the reader reveals any part of the celebrity's name or passes, a point is lost, and another paper is selected. This continues until the minute is up. The papers are not added back to the pool.
- At this point it's another team's turn, and the process is repeated with a new reader and guesser. Each team takes turns this way until all of the papers have been used. The team with the highest total number of correct guesses wins.

Once you are done:

1. Assuming you were creating your own program from scratch to play Celebrity, brainstorm which classes might be used. Make sure to include a class that keeps track of overall game information.

2. Remember instance variables in a class represent information associated with an object (think nouns). Based on your experience playing the game of Celebrity, list what information might be needed in the `Game` class. This list will be improved upon in a later activity.

3. Write up a list of behaviors that might be needed for the `Game` class. What are the things the `Game` class must do? An example behavior would be to play a game, or update a team's score when the guesser correctly names the celebrity.

4. Looking at question 3, which behaviors might you make into methods? Why might you make that behavior a method?

5. Assuming we would like to use a `play` method to organize and call other methods in the `Game` class, describe the `play` method. Because it has not yet been implemented, use pseudocode, a list, or whatever best outlines what the `play` method described above should do.

Check Your Understanding

6. Pick a real-world object and identify the information and behaviors associated with it. Knowing what you do about primitive data, would you consider any of the information needed for the object to also be an object?

7. Share your real-world object with a partner and provide suggestions on the information and behaviors you have chosen. Make adjustments to your own object based on feedback from your partner.
