**AP Computer Science I**

Performance Task

# **Create — Celebrity Lab**

## **Overview**

In the Celebrity Lab students create an application to play a version of the game Celebrity. Celebrity is a charades-like game where each player adds information about several “celebrities” to a pool. A random celebrity from the pool is picked and one player provides clues about the celebrity‘s identity to another player until the other player guesses the celebrity‘s name. To play as a competitive game, teams of two work to guess as many celebrities as possible in rounds of one minute each until the pool is exhausted. The team with the most correct guesses wins. The Java application involves the use of classes and inheritance. A graphical user interface (GUI) is provided that will work for the user interface as long as certain specifications for the other classes are followed. Although this lab uses a GUI, the focus should be on class design, inheritance, and polymorphism. The game provides the context with which to apply these concepts.

Being able to categorize items allows for easier searching and the ability to find related items even if you‘re not sure what it is you‘re looking for. Structure and organization can be found everywhere, from biological specimens to the videos available on streaming services. For computer programs, being able to take advantage of relationships means the potential for less programming time and easier maintenance and enhancement of existing programs.

## **Assessment**

You will be provided with 12 hours of class time to complete and submit the following:

* A video of your program running
* Written responses about your program and development process
* Program Code

Your teacher will share submission guidelines that include suggestions for creating video and PDF files.

## **General Requirements**

You are required to:

* Iteratively design, implement, and test your program.
* Independently create at least one significant part of your program.
* Create a video that displays the running of your program and demonstrates its functionality.
* Write responses to questions about your program.
* Include your entire program code.

## **Program Requirements**

Your program must demonstrate:

* Students are able to create a new class and attempt to create a constructor with the same name as the class, but incorrectly include a return type for the constructor.
* Students are able to write program code to declare instance variables and write headers for constructors with or without parameters, but sometimes fail to assign a value to the instance variables in the constructor. They might unknowingly create local variables in the constructor with the same name as the instance variables instead.
* Students are able to write the correct syntax for a class header with inheritance (extends SuperClass), but have difficulty writing program code to override methods in the superclass because they do not understand that the method signatures must be the same
* Students are able to write program code that calls methods that are in the same class, but have difficulty writing program code to call methods in a superclass.
* Students are able to write program code using some of the data available, but sometimes are unaware of what variables (instance variables / parameters) are available to solve a problem and the associated data that can be accessed by calling the methods of reference data.

## **Submission Requirements**

### 1. **Video**

Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. Your video must not exceed 1 minute in length and must not exceed 30MB in size.

### 2**. Written Responses**

Submit one PDF document in which you respond directly to each prompt. Clearly label your responses 2a – 2e in order. Your response to all prompts combined must not exceed 750 words, exclusive of the Program Code.

## **Program Purpose and Development**

1. Provide a written response or audio narration in your video that:

Identifies the programming language.

* Identifies the purpose of your program.
* Explains what the video illustrates.

(Approximately 150 words)

1. How do we identify if a method is an overridden method? How do we send information from the subclass to the superclass? What keyword is used in Java to identify inheritance? What method is executed when an ArrayList is made of the superclass but a subclass instance is stored in it?

(Approximately 200 words)

1. Why did you choose to implement this program? Describe the development process used in the completion of the project. . Describe another class that could be designed as a subclass to the superclass you created. Describe additional attributes and behaviors for this new class and explain how this subclass would be useful.

(Approximately 200 words)

1. Create the inheritance UML diagram for the classes you created. . Copy and paste one code segment that uses polymorphism. Other than specific syntax, describe how implementing this program without inheritance would change the complexity of your program, using your copied code segment as an example.

(Approximately 200 words)

1. Capture and paste your entire program code into the PDF.

* Include comments or citations for program code that has been written by someone else.

## Tasks

### **Activity 1 - Explore**

**Description**

Students need to understand how to write classes used to construct objects, including instance variables, constructors, accessor methods, mutator methods, and overriding methods from a superclass. While they will need to understand inheritance in a later activity, it’s not necessary for the first three activities of the lab. Below are two links that might be helpful. The first is on object-oriented concepts in general, and the second focuses on constructors. These could be useful as a pre-reading assignment, or as remediation if necessary.

* [https://www.csee.umbc.edu/courses/undergraduate/CMSC331/fall08/0101/ exams/OOConcepts.html](https://www.csee.umbc.edu/courses/undergraduate/CMSC331/fall08/0101/)
* <https://beginnersbook.com/2013/03/constructors-in-java/>
* Plan Project

Time To Complete: 1-3 Hours

### **Activity 2 - Research**

**Description**

In this lesson, students will learn about superclasses and subclasses. A superclass is a parent class that contains common attributes and behaviors used by subclasses (children). Subclasses can draw upon the existing attributes and behaviors of the superclass without repeating these in the code. They will create an inheritance relationship from a subclass to the superclass.

Students will learn about recursion and recursive methods. A recursive method is a method that calls itself. Recursive methods contain at least one base case, which halts the recursion, and at least one recursive call. They will determine the result of executing recursive methods.

* Unit 9
* Unit 10

Time To Complete: 10-11 Hours

### **Activity 3 - Ideate**

**Description**

In this activity students are introduced to the game of Celebrity. Then students will brainstorm different design options for creating a computer version of the game, including the Game class and the play method. Although students are encouraged to brainstorm the classes that make the most sense to them, because they will be utilizing a provided GUI, their focus will be narrowed as they progress through this activity. Spending time talking about different design decisions that they might have made, and how they would affect the GUI and other provided code is a worthwhile discussion that could be had at many points throughout this lab.

For the sake of time, it’s best to try and split the class into groups of six (which will be three teams of two) to play a round. Depending on numbers, it’s possible to have groups of different sizes, however the smallest size group must be four students to have two teams of two in each group. Any group with an odd number will just have one team with more students, and the extra student will act as an additional guesser for their team. Each student should write down five celebrity names.

As teams are playing, ask students to think about the information and behaviors that are taking place. Encourage them to write notes on the classes that will be needed as well as the attributes and behaviors of these classes as they play, which will help with later questions in the activity.

* Introduction to Celebrity

Time To Complete: 2-3 Hours

### **Activity 4 - Evaluate**

**Description**

Students will review and test each other’s code with test cases and provide feedback to the creator on how they might improve their program or to OK the program to move onto the next phase.

This activity focuses on creating a new class. Because students are being asked to design the Celebrity class, it is important to not give them the provided Celebrity.java file until after they have completed question 4. In addition, the LiteratureCelebrity class contains code that could be similar to what students are implementing. It is strongly encouraged to not give students access to LiteratureCelebrity.java until after they have completed this activity. The Focus Alert provides the answer for question 2, so is not included in the student handout.

It is important for students to understand the differences between a constructor and a method, so it is worth reviewing this alert as a class at some point after students have answered question 2. One option would be to include this in the lab debrief.

* The Celebrity Class: A Simple Version

Time To Complete: 1-3 Hours

### **Activity 5 - Construct a Prototype**

**Description**

Although GUIs are not part of the framework for this course, they can be an excellent way to engage students. If there is time and interest, discuss the basic GUI structural components of a Frame (JFrame), Panel (JPanel), and the UI components of buttons, and fields. If this lab is being completed after covering inheritance in Unit 9, incorporate the role of inheritance in the existing GUI libraries. The first question in this activity asks students to explore the official Java tutorial on using swing components to investigate a class that they are interested in. Depending on the depth with which you would like students to learn about GUIs, this question could be skipped or done as a guided practice with the entire class.

The logic of a program is often referred to as the “back-end,” while the GUI is often referred to as the “front-end” because it’s what a user interacts with. Students do not need to know this terminology, but they may have heard it and it might be useful to define it here.

* Putting it all together

Time To Complete: 3-5 Hours

### **Activity 6 - Improve the Design**

**Description**

This activity gives students the chance to design their own subclass of Celebrity. They should spend time thinking about the different attributes and behaviors that they might want to include in this new class. Emphasizing that time spent on the design before beginning implementation can cut down on the time needed to implement a class. Do not provide students with the LiteratureCelebrity class until after they have designed their own subclass of Celebrity.

* Extending the Celebrity Class

Time To Complete: 3-5 Hours

### **Activity 7 - Share Solutions**

**Description**

As a class, spend a few minutes reviewing the requirements of the open-ended activity.

The goal of the activity is to allow students to demonstrate their knowledge of inheritance in a way that is interesting and engaging to them. While it’s possible to create additional constraints or requirements, it’s best to provide students with as much freedom as possible. The use of the Celebrity GUI is intentionally not included in the list of requirements; however, it’s possible that students will continue to work with the given classes from this lab.

* Open-Ended Activity

Time To Complete:1-3 Hours

### **Activity 8 - Reflect**

**Description**

Students answer the reflections questions and submit all evidence.

Time To Complete: 1 Hour