

Application Project Draft 1:

Understanding ArrayList functionality in Java

Nicole Hite

Cato College of Education, University of North Carolina at Charlotte

EIST 6100/8100-085: Foundations of Learning, Design and Technology

Dr. Christa Guilbaud

October 29, 2021

Context Analysis

Public High School (PHS) is a large high school in a suburban area. Many students take online courses through State Virtual Public High School (SVPHS) because of large class sizes and high teacher-to-student ratios at PHS. There is a lack of CS teachers across the United States, and because of the lack of Computer Science (CS) teachers at PHS and surrounding areas, students at PHS who are interested in taking CS classes are encouraged to take the online asynchronous CS courses offered through SVPHS. The CS course curriculum at SVPHS includes the following topics: Primitive Types, Objects, Boolean Expression and “If” Statements, Iteration, Writing Classes, Arrays, ArrayLists, 2D Arrays, Inheritance, and Recursion.

I will create a **short tutorial** that teaches students how to declare, initialize, access, and modify ArrayLists in Java. Students can use any Java programming environment of their choice to code their program. For my demonstration, I will be using Apache Netbeans 8.2 for the programming, and Google Slides for the visual explanation. I plan for this video to be around 5 minutes because the content of this topic should be fairly straightforward at this point in the course given the student’s previous knowledge at this point in the course but will be sure to clarify any necessary components that will increase understanding.

Learner Analysis

The learners for the tutorial on Arrays in Java are students in a first-year introductory computer science asynchronous online course. The students taking this CS course will be enrolled in SVPHS and should have a device capable of completing this virtual course, capable of programming in Java, and capable of watching online videos. The students in an introductory computer science course are still developing computational thinking and algorithmic thinking skills. Prior to learning about ArrayLists in Java, students should already have a basic

understanding of Java data types, basic syntax, arithmetic operators, looping, variables, and Arrays to assist them in successfully comprehending the tutorial. Students will need to understand ArrayLists because ArrayLists are a common data structure utilized in programming, and are a foundation for implementing important, complex, and commonly used data structures such as Queues and Stacks in higher level programming courses.

ABCD Objective & Blooms category:

Goal: Students will learn about ArrayLists in Java.

Objective: By the end of this eLesson, introductory computer science students will be able to declare, initialize, access, and modify ArrayLists in Java with objects of different types.

Bloom's Taxonomy: This objective can be classified under **Apply**. I believe this objective is classified under **Apply** because the student will first have to understand the basic characteristics of ArrayLists, then they will have to use their new knowledge of ArrayLists to create ArrayLists that objects of different types such as Integers, Doubles, Strings, custom class objects, etc.. This encourages students to apply what they learned to demonstrate their understanding of the topic.

Audience	By the end of this eLesson, <u>introductory computer science students</u> will be able to declare, initialize, access, and modify ArrayLists in Java with objects of different types.
Behavior	By the end of this eLesson, introductory computer science students will be able to <u>declare, initialize, access, and modify ArrayLists in Java</u> with objects of different types.

Condition	<u>By the end of this eLesson</u> , introductory computer science students will be able to declare, initialize, access, and modify ArrayLists in Java with objects of different types.
Degree	By the end of this eLesson, introductory computer science students will be able to declare, initialize, access, and modify ArrayLists in Java <u>with objects of different types</u> .

Learning Theory

The Learning Theory I intend to adopt for this scenario will be **Cognitivism**. I am using the Cognitivist Learning Theory because I want students to use concepts they should already know of at this point in this course such as wrapper classes, primitive data types, methods, object instantiation, and object-oriented programming to further support their understanding of ArrayLists. Students will recall information to process and make connections of the old information and new information, and mentally organize the new information about ArrayLists in their brain.

Instructional Design Model

The Instructional Design Model I intend on applying for the design of my instruction is **ADDIE**. I will use the ADDIE model because it is an iterative model with a thorough, structured, and straightforward planning process that encourages improvement based on feedback from learners. The ADDIE model is appropriate for this instruction because the Analysis, Design, and Development of this instruction has to be done prior to the Implementation because it is in a video format. Once the video is produced, I will distribute the video. Making changes to a video mid-way through the Implementation phase can be complex, so waiting until after the implementation of the video (the Evaluation phase) to make changes would be beneficial. This is

where the ADDIE model recycles, and the Analysis, Design, and Development phase starts again.

Analysis

In this phase, I identified an objective and goal for the CS students. I considered the characteristics of the students along with their existing knowledge at this point in the course when the content for ArrayLists is taught, along with the computational skills they are developing. I also considered what technology they should have access to. I will research existing tutorials on ArrayLists and decide what ways those videos could be improved. I will also do research on best practices for CS education to see if there is any applicable advice for creating CS-related tutorial videos.

Design

In the Design phase, I will think of different scenarios I can utilize as an example for ArrayLists to improve understanding of the fundamentals of this data structure. I will need to decide how many examples would be necessary, and how long each example should be. I will also sketch an idea of what diagrams and flowcharts I need to include in this video to benefit the learners comprehension of the content. I will also make a prioritized checklist based on what specific topics need to be included, and how important they are to this topic to ensure I do not stray from my goal and objective.

Development

In the development phase, I will program the first draft of the tutorial that I will do a demonstration & live-coding session with during the recording. I will write out my script. I will then record the live-coding tutorial and explanation, edit the recording accordingly, and upload the recording. The tools I have accessible to me that I prefer for

video development are Open Broadcaster Software (OBS) Studio for recording, and Adobe After Effects for editing. I plan to upload the completed video privately to YouTube. This allows for the video to be played anywhere with internet access and allows for multiple videos to be grouped together in one playlist if I decided to make numerous videos in a series.

Implementation

The Implementation phase for this project will not be utilized in a classroom setting, as I am not an instructor for a CS course. Instead, I will post the video to YouTube with a private link. I will share this video with 2 groups of people: 1) Peers I know who are currently introductory CS students, and 2) Peers who have graduated with CS degrees who know Java.

In a real-world Implementation of this content, I would provide the link to my students during their weekly content module. I would include links/downloads to the files I worked on in the live-coding tutorial so they can play with the code by changing it to see how their changes impact the outcome of the code. This would be beneficial for students who learn by “Trial-and-Error” and manipulation.

Evaluation

I plan to share this video with 2 different groups to get feedback from novice students with a surface-level understanding of Java (the target audience) to understand how I can improve the video for their understanding, along with professionals in Java so they can provide feedback to ensure the terminology and technical explanations are correct. Both perspectives will ensure that I am providing accurate and effective tutorial content.

In a real-world setting, I would use assessment techniques such as a survey or quiz to see if students in this class appear to understand the content better than the previous semester/year. Instructional design is all about adapting to the needs of the learners, and it can be a long-term process to create a good learning tool. I would take this feedback, and continue to cycle through the ADDIE model again.