**AP Computer Science I**

Performance Task

# **Create — Interactive Resume**

## **Overview**

In this performance task students will learn to make a resume. Students learn about website layouts, and how to construct a website so that it is aesthetically pleasing. Students will use these concepts to evaluate the quality of formatted resumes. Students will draft their own interactive resume and make changes based on constructive feedback and link to their portfolio page. They will learn about hosting a website and how the internet works at a basic level.

## **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:

* Infer the usage of compound boolean Expressions.
* Create programs that include sequences, events, loops, and conditionals.
* Create basic events (user input/buttons) to utilize user input in programs.
* Break down problems into smaller components through systematic analysis, using constructs such as procedures, methods or functions.
* Create clearly named variables that represent different data types.
* Explain how data structures (e.g., numbers and strings) work and how they are used to create algorithms.
* Document different types of errors which occur in code, and fix errors with little teacher assistance.
* Explain why validation is important to program design and include some exceptions in programs.
* Justify practices and tools for debugging, including code debuggers, unit testing, test suites, and continuous integration.
* Use debugging strategies to correct and prevent errors in code.
* Justify the design decisions they or their team made in creating a program in a way that allows others to follow and understand.

## **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. Infer the usage of compound boolean Expressions in your program. How did you use sequences, events, loops, and conditionals in your program? How did you include basic events to utilize user input in your program?

(Approximately 200 words)

1. How did you break down problems into smaller components through systematic analysis, using constructs such as procedures, methods or functions. How did you use functions with parameters to organize code and make it easier to reuse?

(Approximately 200 words)

1. How did you create clearly named variables that represent different data types? Explain how data structures (e.g., numbers and strings) work and how they are used to create algorithms

(Approximately 200 words)

1. Document different types of errors which occur in code, and fix errors with little teacher assistance. Explain why validation is important to program design and include some exceptions in programs.

(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**

What is a resume? How do we write a resume? Students will explore an example resume and watch a tedTalk on what to include in a resume and Student’s will generate a list of personal things to include in their resume.

* What is a Resume
* Web Theory and Layout

Time To Complete: 1 Hour

### **Activity 2 - Research**

**Description**

This series digs into the things you need to write larger and larger applications with JavaScript. This is where it gets real! Let's look at a few more practical applications of JavaScript and learn about a few useful tools that are widely used in the industry.

* Fundamentals Part 1
* Fundamentals Part 2
* JavaScript Developer Tools
* Fundamentals Part 3
* Problem Solving
* Understanding Errors
* Project: Rock Paper Scissors
* Learning Layouts
* Sample Interactive Layouts

Time To Complete: 12-15 Hours

### **Activity 3 - Ideate**

**Description**

In this unit, you will draft and build an interactive resume. Before we do that, we need to learn a little more about what a resume is, why you need one, and what goes into creating one.

* Clean Code
* Fundamentals Part 4
* DOM manipulation
* Project: Etch-a-Sketch
* Draft Interactive Resume

Time To Complete: 3-5 Hours

### **Activity 4 - Evaluate**

**Description**

Students will view and critically evaluate each other’s resume and digital resume. They will provide feedback and plan for improvements for both resumes.

Time To Complete: 2-3 Hours

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

**Description**

Students will develop a final look of their resume design. They will have a working web page hosting a digital resume and a downloadable version of their attached to the website. Along with finalizing the last of the JavaScript projects which can be used as work samples on their portfolio/resumes.

* 12.Fundamentals Part 5
* 13. Project: Calculator

Time To Complete: 3-5 Hours

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

**Description**

How can you improve your design? Put your solution to the test, and seek and incorporate feedback into your design.

Time To Complete: 1-2 Hours

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

**Description**

What is the result of your design process? Present your final product and process to your peers, including information about each stage of the design process.

Time To Complete:1-3 Hours

### **Activity 8 - Reflect**

**Description**

What did you learn through the engineering design process that could be helpful to others? Reflect on your design process and work. After the presentation of the project, reflect on the process from beginning to end as well as yourself as a learner.

Time To Complete: 1 Hour