Create Performance Task Checklist

**What you do**

Develop a program that demonstrates:

- Use of several effectively integrated mathematical and logical concepts, from the language you are using
- Implementation of an algorithm that integrates other algorithms and integrates mathematical and/or logical concepts
- Development and use of abstractions to manage the complexity of your program (e.g., procedures; abstractions provided by the programming language; APIs)

Collaborating with another student is not required, but is strongly recommended.

**What you Submit**

- A video of your program running. *Your video must not exceed 1 minute in length and must not exceed 30MB in size.*
- Written responses about your program and development process
  - Program Purpose and Development (~150 words)
    - Programming language
    - Purpose of the program
    - Explanation of what's illustrated in the video
  - Description of the incremental and iterative development process of your program, focusing on two distinct points in that process. (~200 words)
    - Describe the difficulties and/or opportunities encountered at two points in the process of developing your program.
    - In your description clearly indicate whether the development described was collaborative or independent.
    - At least one of these points must refer to independent program development; the second could refer to either collaborative or independent program development.
  - Code segment that implements an algorithm (marked with an oval in the printed code) that is fundamental for your program to achieve its intended purpose. (~200 words)
    - Must include an algorithm that integrates other algorithms and integrates mathematical and/or logical concepts.
    - Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program.
  - Code segment that contains an abstraction you developed (marked with a rectangle in the printed code). (~200 words)
    - Your abstraction should integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program.
- Program Code
  - A .pdf of your code
  - Comments or citations for program code that has been written by someone else.