Batch Scheduling Assignment

Question: Consider 5 jobs, A through E, with runtimes 3, 5, 2, 2, 2 and arrival times 0, 0, 5, 5, 5 respectively.

Task: **1. Source Code (30 points):**

* Provide an implementation to compute the **average wait time using** **Shortest job first and Running them in a defined order (B, C, D, E, A )** using Python**,**where the user provides the value for runtimes and arrival times **via the keyboard (**do not manually hardcode runtime and arrival times values**)**.
  + - The program should start by asking user the number of jobs to be executed i.e., a user-friendly program.
* **Output:**Running program should display the **average wait**for the shortest job first and running them in the order (B, C, D, E, A) in a user-friendly form.
  + - *Extra point:* Running program should display the **average wait**for the shortest job first and running them in an **order generated randomly.**Extra credit will ONLY be considered if the required initial functionality is completely implemented.
* **Points Distribution**:
  + - Coding style and clarity, 5 points.  
                 ---->Appropriate parenthesis locations, indention, etc.   
                 ---->Always write comments at the beginning of source files   
                                 Header comments: Author, date, history, etc.  
                 ---->Always write comments at the beginning of a non-trivial class/function.  
                                 What this class/function does, high-level algorithm if needed.  
                 ---->Write in-line comments for non-trivial functions/blocks/lines of code.
    - Functionality, 25 points

**2. READMe file (5 points):**A design document that clarifies the followings:  
                            ---->At a high level, how you implement those required functionalities.  
                                                   E.g., any edge cases your program has considered.                                    Think of a **.txt file** that states the step involved in successful execution of your code.

3. **Pdf document (10 points)**: Provide a visual representation of both cases showing the runtimes and arrival times as in Example 1 on slide 12 of today's class [ch-02e.pdf](https://lincolnu.instructure.com/courses/10509/files/940316?wrap=1)[Download ch-02e.pdf](https://lincolnu.instructure.com/courses/10509/files/940316/download?download_frd=1).

**Submission:**

This is an individual assignment.

Upload your **solution as a single zipped file (.rar, .zip)**that contains the source code (*with the appropriate file extension like .java, .py, .c, .c++*), READMe file / Design document and pdf document. Non-adherence to instructions would result in losing points.