***INMEDIA – Activity 3***

**Project Description**

|  |  |
| --- | --- |
|  | DESCRIPTION |
| Project Title | Linked Lists, Initializations, and Operations |
| Target Audience | Programmers and developers interested in data structures and file operations. |
| Purpose and Need | The "Linked Lists, Initialization, and Operations" program is designed to read child profile data from an input file ("INPUT.TXT"), create a linked list of child profiles, sort the list based on last names and first names, determine gifts for each child according to specific criteria, and write the child profiles along with their assigned gifts to an output file ("OUTPUT.TXT"). |
| Overall App Objectives | Implement a linked list data structure to efficiently manage child profiles.  Read child profile data from an input file and construct a linked list.  Sort the linked list in ascending order based on last names and first names.  Determine appropriate gifts for each child based on age, gender, and good/bad deeds.  Write the sorted child profiles and their assigned gifts to an output file.  Provide clean and maintainable code with well-structured classes and functions. |
| Technology used to build the application (e.g. Android/iOS) | Pycharm |
| Budget (min, average, high budget) | Minimum |
| Programming Language used to develop software | Python |
| Software Features | The program comprises several essential functions and classes:  lastname\_mp3\_classes.py  This module contains class and function definitions used in the main program.  It includes the ChildProfile class representing child profiles with attributes such as first name, last name, gender, age, good deeds, bad deeds, and gift.  The Node class represents a node in the linked list.  The ChildProfileList class represents the linked list of child profiles.  build\_list\_from\_file(self)  A method of the ChildProfileList class responsible for reading data from the "INPUT.TXT" file and constructing the linked list of child profiles.  swap(self, node\_1, node\_2)  A method of the ChildProfileList class that swaps the data of two nodes in the linked list.  sort\_list(self)  A method of the ChildProfileList class that sorts the linked list in ascending order based on last names and, if necessary, first names using the bubble sort algorithm.  determine\_gift(self)  A method of the ChildProfileList class that assigns gifts to each child profile based on age, gender, and good/bad deeds.  write\_gift\_file(self)  A method of the ChildProfileList class that writes the child profiles and their assigned gifts to the "OUTPUT.TXT" file.  lastname\_main.py  The main program file that orchestrates the entire process, including importing necessary classes, creating instances, and calling methods to execute the program logic. |
| Design Strategy | No design since it’s a machine program (console-based) |
| Review and Approval | The project has been reviewed and approved, meeting all the requirements specified for building a linked list, initializing it with child profile data, performing sorting and gift assignment, and writing the results to an output file. |