|  |  |  |  |
| --- | --- | --- | --- |
| **CodeAIR Remix Mastery Rubric** | | | |
| **Requirement** | **No evidence ←—-------------------------------------------------------→ Mastery** | | |
| **Programming Conventions** are followed | * Variable names aren’t descriptive * Function names aren’t descriptive * Code blocks inconsistently indented * Capital letters used * Code is not organized into sections |  | * Variable names are descriptive * Function names are descriptive * Code blocks consistently indented * Use of lower-case letters and underscores * Code is organized into sections |
| **Documentation and Readability** | * No comments are used. * Code is difficult to read because no blank lines were used, or too many blank lines were included. |  | * Frequent and descriptive comments are used regularly. * Blank lines are used to help with readability. |
| **Use of Variables and constants** | * “Magic Numbers” or literal values are used in the code. * Data isn’t tracked or updated (no counters, states, conversions, etc.). |  | * Constants are used to eliminate “magic numbers.” * Variables are used for storing, keeping track of and updating data. * Global and local variables are used. |
| **Use of Functions** | * No plan or algorithm to follow. * Everything in one main program. * Long sections of code. * Functions use all global or all local variables (no use of parameters or return) |  | * Code is divided into smaller sections that accomplish a task. * Parameters are used as needed. * Local and global variables are used as needed. * Functions return a value as needed. |
| **Use of libraries and modules** | * No libraries are imported. * No custom modules are imported. |  | * Public libraries are imported, and functions used. * Custom modules are imported, and functions used. |
| **Use of Inputs**  Buttons and sensors | * No input during program run. (from sensors or buttons) |  | * Input is used during program run. (from sensors or buttons) |
| **Algorithms and Programming** | * No algorithms identified or used. * Program does not utilize input * Data structures are not used to simplify code.  (such as lists and tuples) * Debugging practices are not used and code contains errors. |  | * Algorithms are used to manipulate data and get results. * Data is used to inform decisions. * Data structures are used to simplify data collection and implementation. (such as lists, tuples, etc) * Debugging practices are used to correct errors in code and logic. |
| **Control Structures** | * Program does not have any branching statements. (if or if/else or if/elif/else) * Program does not use any iteration. (while or for loops) |  | * While loops and if statements are used to control the flow of execution. * Conditional and logical operators are used appropriately. * Nested while, if statements used when needed. |
| **Use of Outputs**  LEDs, speaker, motors | * No output is produced; nothing happens. * Console output is unstructured. |  | * One or more outputs are used to convey data or perform a task. (motors, lights, speakers, etc.) * Console output is formatted. |
| **Collaboration** | * Students work independently or uncooperatively on a team. |  | * Students work collaboratively with shared tasks in their team to complete the project. |
| **Synthesis / Purpose** | * No clear purpose for the program. * Project does not incorporate learning across the mission pack. |  | * Program’s purpose is clearly understood. * Project combines learning, concepts and code from several missions. |
| **Code Completion** | * Code will not run or doesn’t complete the task correctly. |  | * Code runs and accomplishes its task without any errors, including logic. |