Instructions and Commands for using Programmophone

General Utilities

- By default, when Programmophone is started, Misty Mode is turned on in which no command will be executed until the keywords' **hey misty'** are spoken along with the regular commands. Meant to be used when at a noisy place or to activate the voice mode without the use of any physical device such as a keyboard or mouse
- Whenever something is inserted into the editor window, a beep sound would be heard to tell the user that the changes are made with the editor.
- The commands for activating and deactivating voice mode are as follows.
    - **Activate voice mode**
    - **Deactivate voice mode**
- For opening any file, the command is **'Open a file',** and then the speech engine calls out the name of each file and waits for the user to respond with **'this one'** or **'this file'.**
- For saving any file, the command is **'Save this file'**. After the command is invoked, say the filename after the chime and wait for further instructions for the file format by the speech engine.
- Say **'Compile this file'** to compile the program, file format is automatically detected.
- Say **'Run this program'** to run the program.

Cursor and Deletion Utilities

- For moving to a specific position, the command is **'Move to line <x> and column <y>',** which could also be used as follows**.**
    - **Move to line <x>**
    - **Move to column <y>**
- For changing cursor position relative to the current position of the cursor, commands are as follows.
    - **Move <left/right/up/down> <optional argument x>**
    - **If <x> is not given, default value 1 is taken; rest is self-explanatory**.
- The command for getting the current cursor position is **'Tell current position'** or **'Tell cursor position'.**
- For clearing the editor, say **'Remove all'**
- Say **'enter'** to do the same as clicking enter button on the keyboard would do, i.e., add a new line

- Say **'escape'** to escape from the region defined by curly braces '{ .... }', used primarily in languages like C/C++ and JAVA. Invoking this command would move the cursor from anywhere inside the block to the right of '}' character.
- For reading a particular line x, one could say **'Read line <x>'** followed by optional argument **<character by character>**. If character by character is used errors due to semi-colon-, brackets, operators could be easily caught. So one will have to say read line <x> character by character for reading every character at a slower pace.

Programming Utilities

C/C++ specific

- For including the header file in C or C++ program, say **'include - '** followed by the name of the header file. For example - include iostream
- For declaring any variable use the following command and format -> **'Declare <datatype> <variable name>'**. For example - long long x
- For adding main function say **'<datatype> main'** to add the main function.
- The commands for adding if, else if, else, while, do-while blocks are following.
    - **Add an if statement**.
    - **Add an else statement**.
    - **Add an else if statement**.
    - **Add a while loop**
    - **Add a do-while loop**.
- A common guideline for adding a condition to if, else if, while, do-while blocks is -> Wait for the chime and then start speaking the condition which is to be added
- For adding a for loop, the command is **'Add a for loop'.** A for loop has three parts Initialization, Condition, and Updation. One needs to tell each of the 3 when enquired, which is just a second after the previous command is executed.

Only C++ specific

- For using namespaces, the command is **'using namespace <ns>'.** For example - using namespace std.
- For printing to the output stream, say **'print - '** followed by anything you want to print to the stream. Note here that in order to print a string, one would need to specify the double-quotes.
- For adding a newline into the output stream, the command is **'Add a newline'.**
- For taking input, say **'Input - '** followed by the variables one needs to scan

The very basic command

- Just say **'Add - '** and everything one would say after Add will be inserted into the editor.

Currently, utilities for adding header files, if-else statements, loops, etc. have been added only for C++ and C (due to similarities with C++) due to time constraints; therefore, language-specific commands will not work while working with files that are not of type '.c' and '.cpp'. However, all other functions discussed under **General Utilities** and **Cursor and Deletion Utilities** would still work.

Also, anything other than the basic building blocks mentioned as C++ utilities would have to be added by using Add command.

Moreover, while the utilities for other languages that are Java, Python, (More in C) aren't added if one needs to use or test our application for these languages, he/she should use Add command as a replacement.

P.S - Note that turning off the voice mode will clear the activity log because the activity log can only keep track of changes made with speech

Sample program #1

Printing numbers from 1 to 100

- Include iostream
- using namespace std
- int main
- Declare int i
- Add a for loop
    - i equal to 1
    - i less than 101
    - i plus plus
- print i

And then Say 'Compile' or 'Run' for running the program.