# Multi-Purpose Programming Language (MPPL or Mapple) Roadmap & Identity Specification (v0.1)

### **1. Purpose / Use Case**

* Learn and gain a deeper understanding of how programming languages work.
* Create an easy-to-read general-purpose scripting language inspired by Python and Java.
* Beginner-friendly but strict enough to encourage proper coding habits.
* **v0.1 Goal:** Teach programming basics, variables, arithmetic, and printing to terminal.

### **2. Syntax Style**

* Semi-strict syntax.
* Statements must end with a semicolon (;).
* Classes end with double semicolons (;;).
* Functions declared using func keyword inside classes.
* Example class and function declaration:

class ClassName::  
 func FunName:  
 let bool isItWorking = true;  
   
 ;  
;;

### **3. Runtime / Execution Model**

* Compiled language.
* Garbage collection not implemented in v0.1 (will be added in future versions).
* **Performance Considerations:**
  + CPU efficiency: compiled to machine code for speed.
  + Memory efficiency: minimal runtime overhead in v0.1.
* **Machine Code vs Bytecode / VM (Still deciding compilation type):**
  + Machine code: direct CPU instructions, faster, less portable.
  + Bytecode + VM: portable, safer, easier to extend, but slower.
* v0.1 will compile directly to machine code for simplicity. (default for now)

### **4. Core Features**

* Supports multiple variable types: int, num, str, char, bool.
* Input/output: input() and print() function.
* Arithmetic: + - \* / %.
* Semantic rules: type consistency, declaration before use.
* **Global scripts:** Code outside functions/classes executes immediately.

let int x = input().int;  
print(x + 2); // executes immediately

//calling input() lets you input a value from CLI (command line interface), you could add a string comment which will be printed in the prompt. Input is only used when initializing or assigning a value to a variable. You specify the data type being assigned when inputting by using “ .<data type> “behind input().

/\* Example:

let str name = input(“Enter your name”).str ;

\*/

In cases like:

print("Score: " + 100);

The program will throw an error in version 0.1.

### **5. Standard Library & Extensibility**

* Minimal standard library in v0.1: input(), print(), basic math functions(addition, subtraction, division and multiplication).
* Open-source model: users can submit libraries to a public repository.
* Installer will provide core language + standard libraries; optional libraries can be added later.
* Future goal: expand to frameworks for web development and low-level systems programming.
* Error Handling: built in error handling for input function for when a wrong data type is entered for the data type that the value is being assigned to. Error handling for print function is available, especially in cases were user would try to use implicit type conversion.

### **6. Data Types**

| Type | Description | Example |
| --- | --- | --- |
| int | Whole numbers | 1, 3682, 931781 |
| num  char | Real numbers (float)  Single character | 1.23, -128.45, 0.001  ‘m’ ‘#’ ‘C’ |
| str | Strings is an array of characters | “Mukona”, “Hello World” |
| bool | Boolean values | true, false |

### **7. Variable Declaration & Initialization**

Type declaration is mandatory when creating a variable.

Implicit type conversion is not possible in version 0.1 but explicit type conversion is possible for the input function.

Semicolons (;) are mandatory at end of lines (and in later versions at the end of control statement and loops).

Double semicolons (;;) will be mandatory at the end of a class.

* Declaration: let <type> <name>;
* Initialization: let <type> <name> = <value>;
* Example:

let int age = 20;

let char initial = ‘C’;  
let str name = "Mukona";  
let num average = 92.9;  
let bool isActive = true;

* Assignment after declaration:

age = 25;  
average = 62.3;  
isActive = false;

### **8. Comments**

* Single-line: // comment
* Multi-line: /\* comment \*/

### **9. Example Program**

let int age = 20;  
let str name = "Mukona";  
let num average = 54.9;  
let bool isActive = true;  
  
print(name);  
print(age + 5);  
print(average);  
print(isActive);

### **10. Identity Summary (v0.1)**

* Purpose: Teach beginners programming basics with clear syntax.
* Syntax: Semi-strict, Python/Java-inspired.
* Execution: Compiled to machine code, garbage collection deferred.
* Core Features: Variables, arithmetic, print, input, global scripts.
* Library Model: Minimal standard library, expandable via open-source contributions.
* Use Case: Beginner-friendly scripting and experimentation with programming fundamentals.