



<b>Document Type</b>	User Manual
<b>Name (Student Number)</b>	<ul style="list-style-type: none"><li>• Ala Al Din Afana (16395986)</li><li>• Shanan Lynch(14723071)</li></ul>
<b>Supervisor</b>	David Sinclair

# 1.Introduction

This document is designed to give an overview of the Mona programming language and how to use the online platform in order to compile and run the code. We assume that readers have a basic understanding of programming fundamentals and are familiar in the basic concepts of either python, java or both.

## 1.1 What is Mona ?

Mona is a programming language designed to help programmers learn the basic concepts of programming. It is integrated into a web application that provides a user friendly UI which is hosted on an EC2 instance to enable users with minimal programming experience easy access and start writing code without the need to worry about installation, dependencies and setting up paths in their local environment. The language is composed of a readable grammar and strict syntax rules to allow programmers to develop clear understandable code building up their programming knowledge over time.

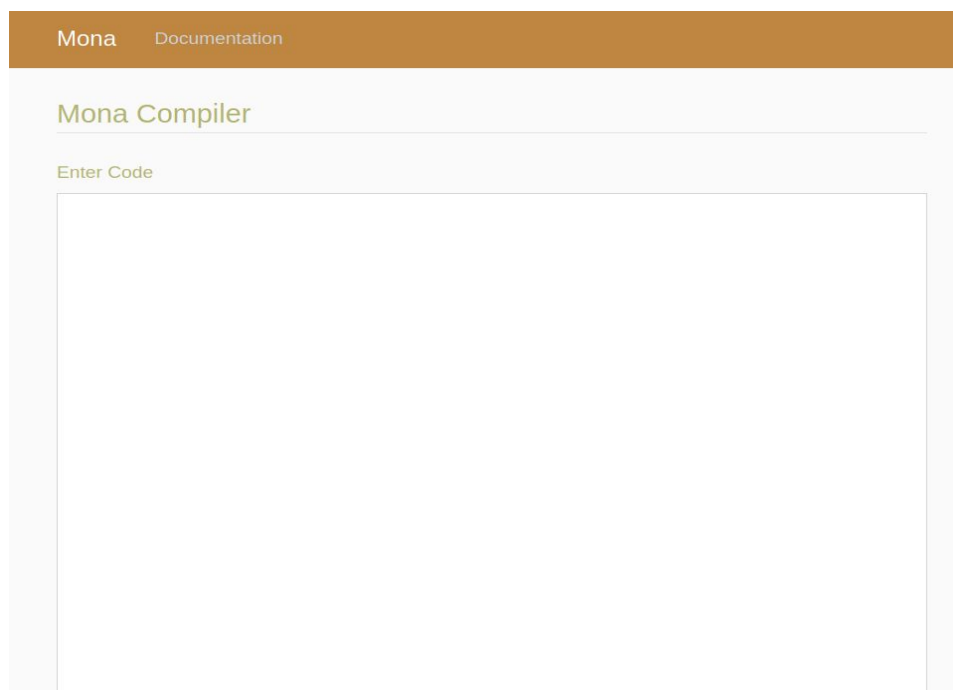
We studied and researched different aspects that make a programming language readable, difficult to transition to (from other programming languages) and to from (to other programming languages). The main goal is to create a language that teaches a programmer how to code without limiting their ability to transition into other languages.

## 2.Accessing the Platform

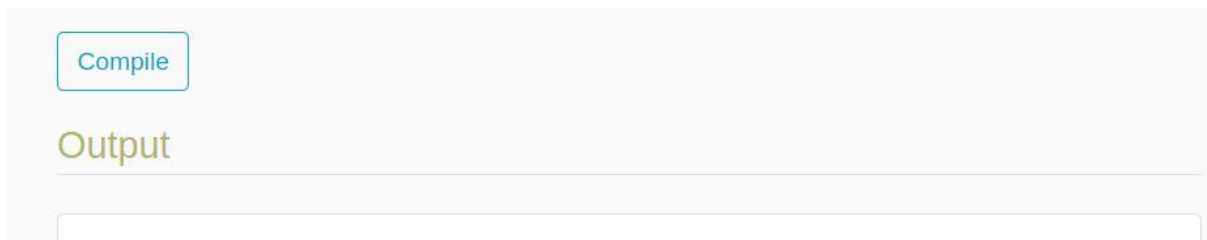
Mona can be accessed by navigating to the following ip address 52.211.181.164.

**Note:** The goal was to buy a domain which will be mapped to the referenced ip address by creating a DNS record in route 53 service but we ran out of time.

### 2.1 UI overview - How to use



- Code can be written in the box under **“enter code” (Above)**



A screenshot of a code editor interface. At the top, there is a light blue button labeled 'Compile'. Below it, the word 'Output' is displayed in a large, bold, black font. Underneath 'Output' is a large, empty rectangular box for displaying the results of the compilation.

- Once the developer is finished writing the code the **“compile”** button compiles the code (**below**)

### ***Hello world example***



A screenshot of a code editor interface. The top section is labeled 'Enter Code' in a large, bold, black font. Below this label is a text area containing the following C code:

```
main{  
    print("hello");  
}
```



A screenshot of a code editor interface. At the top, there is a light blue button labeled 'Compile'. Below it, the word 'Output' is displayed in a large, bold, black font. Underneath 'Output' is a large rectangular box containing the text 'hello'.

For more details regarding grammar and syntax please refer to section 3.

## **3. Grammar and Syntax**

### ***3.1 Overall Program structure***

Global variables followed by a list of functions , followed by main function

### 3.2 Identifier types

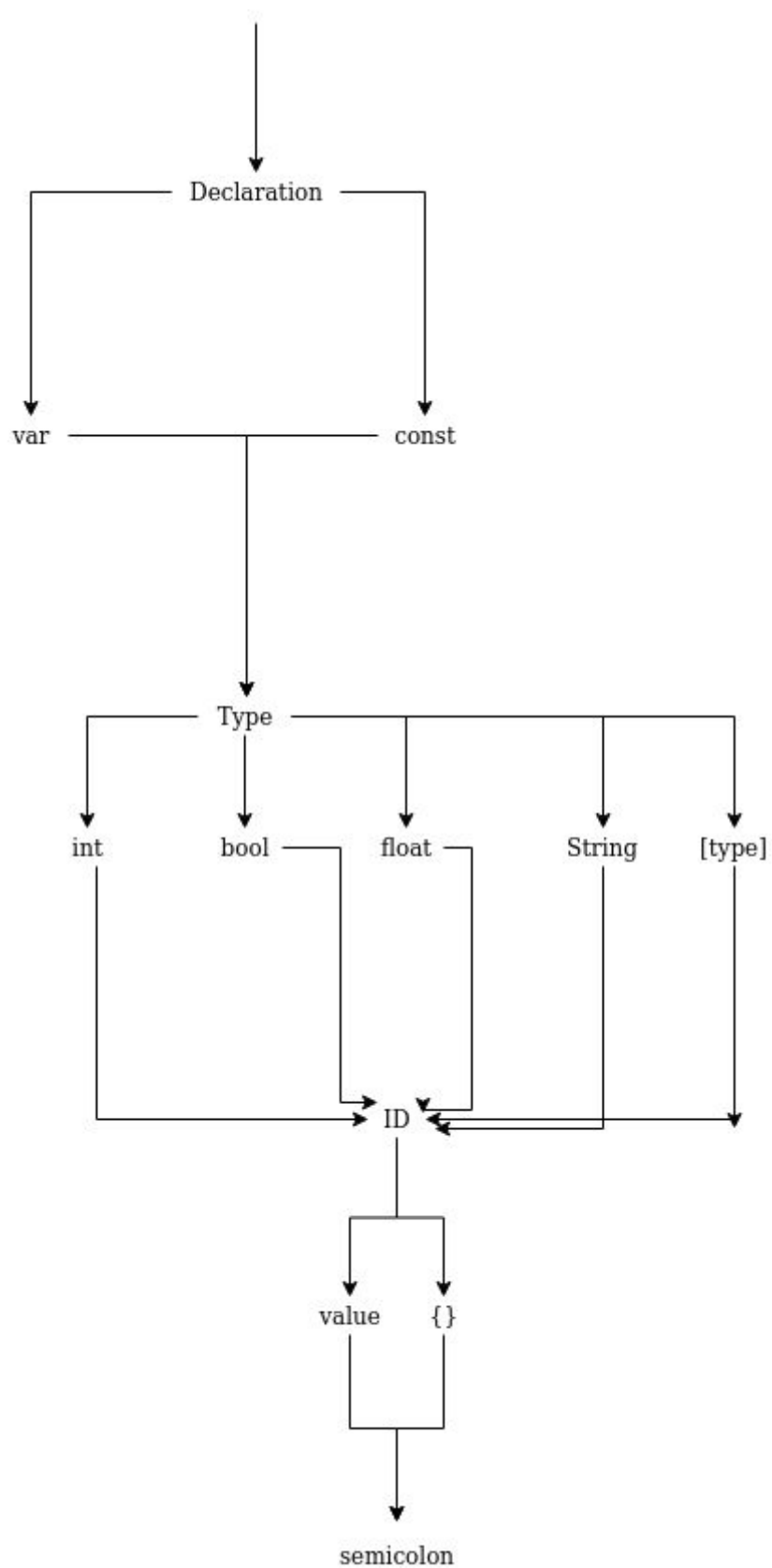
- **Var**
  - Identifiers declared as var are variable types
  - These variables can be reassigned multiple times
- **Const**
  - Identifiers declared as const cannot be re assigned
  - Attempts in reassigning them will lead to a semantic error

### 3.3 variable types

- **Int**
  - Variable type ints are Integer variable with any combination of any whole numbers that have no decimal points
  - Eg. 1,2,3,4,
- **Float**
  - Float types are any numbers with a decimal point . whole numbers cannot be assigned to this data type
  - Eg. 1.002
- **Bool**
  - Bool can be assigned boolean variables such as true or false
- **string**
  - strings are more complex data types they contain any combination of ascii characters
  - A string is a list of characters
  - When strings are declared they are quoted.
    - Note string does not start with a capital s
  - E.g "hello world"
- **[type]**
  - Arrays are declared by "[" + any of the above types followed by a "]"
  - Eg [1,2,3,4,5]

*[Continued next page]*

### 3.4 variable declaration map



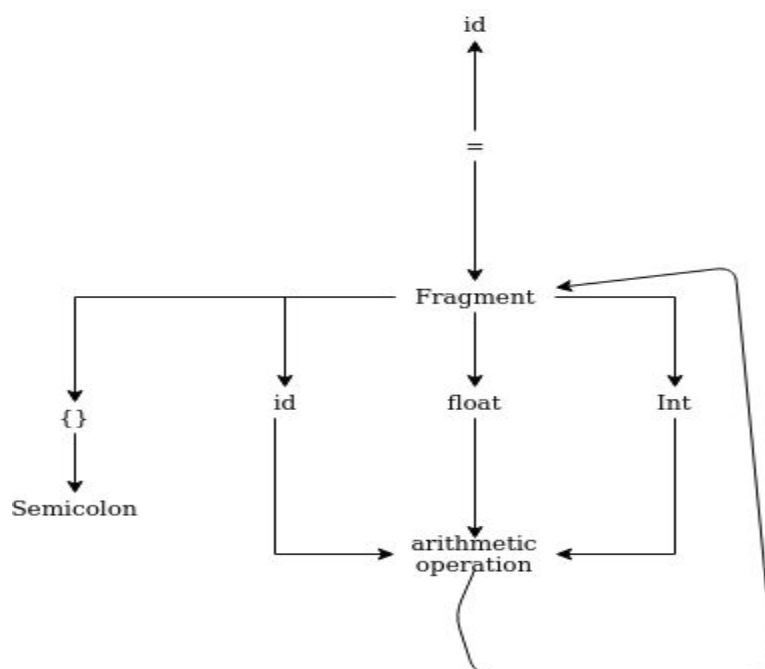
- Example `const int a = 0 ;`

### 3.5 Arithmetic operations

## 1. Arithmetic operations

- follow the Bomdas rules ( brackets order multiplication division addition subtraction)
- Can be performed on int and floats
  - +
    - Addition between two values
    - Eg.  $1 + 1 = 2$
  - -
    - Subtraction between two numbers
    - $1 - 1 = 0$
  - \*
    - Multiplying two numbers together
    - Example  $2 * 2 = 4$
  - /
    - Division dividing two values with each other
    - $2/2 = 1$
  - ^
    - One value to the power of another
    - $2^2 = 4$
  - %
    - Modulus
    - $2 \% 2 = 0$
- You can assign the values of arithmetic compilation to a variable.
- You can perform arithmetic operations on variables ints and floats as long as LHS and RHS of operation are of the same type

### 3.6 Arithmetic operations map



### 3.7 conditional statements

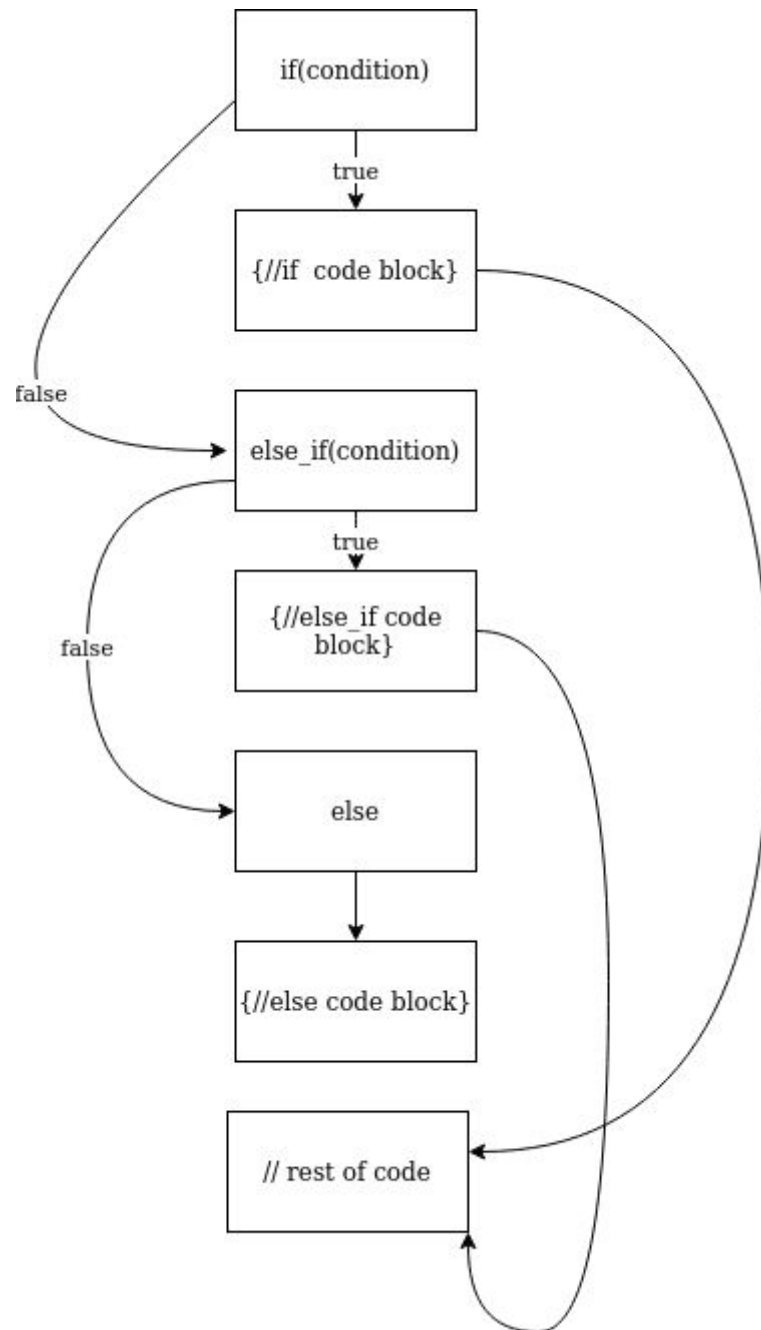
- Conditional statements compare two fragments with a comparative operators
  - ==
    - Equals to operator true if the lhs and rhs are equal else false
  - <
    - Less than operator : true if lhs is less than rhs else false
  - <=
    - Less than equal or equal to operator : true if lhs is less than or equal rhs else false
  - >
    - Greater than operator true if lhs is greater than rhs else false
  - >=
    - Greater than operator true if lhs is greater than or equal to rhs else false
  - !=
    - Not equal operator True if lhs is not equal to rhs else false
  - &&
    - Logical And operator
    - If both statements on lhs and rhs are true evaluate as true
    - Else false
  - ||
    - Logical Or operator
    - If either lhs or rhs or both are true evaluate to true else false

### 3.8 If ,else\_if and else statements

- If else\_if and else are control flow statements
  - **If** statements has a conditional statement if evaluated to true the body of the if statement is executed after execution is finished the flow of the code leaves the block and execution is continued
  - **Else\_if** statements can follow if statement, if the if statement evaluated to false the control flow is send to the else\_if statement and checks what the else\_if evaluates to , if it evaluates to true the code block is executed if false the control flow goes to the next statements
  - **Else** statements follow either ifs or else\_ifs. The code block in else statements is evaluated if the if statement and else\_if statements all evaluate to false
- **Note:** Else\_if can be followed by multiple else\_if statements or an else statement

*[Continued next page]*

### 3.9 if, else\_if and else Control Flow Diagrams



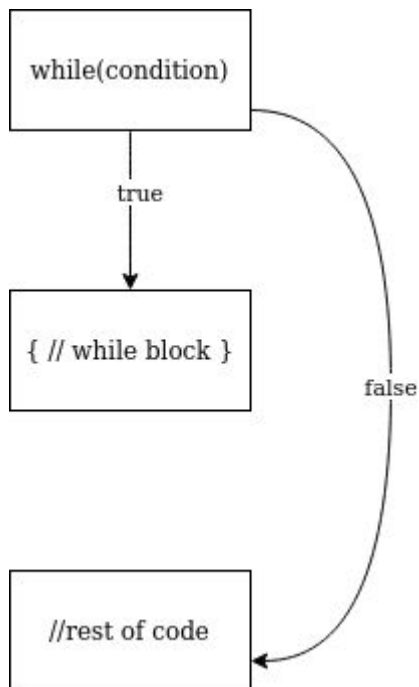
### 3.10 while loops

- While loops take conditional statements and execute the same block of code until the statements is false

*[Continued next page]*



### 3.11 while loop control flow diagram



### 3.12 Built in functions

#### String

- `.getChar(index)`
  - Built in function gets a character at index position
- `.length()`
  - Gets length of a string

#### Array

- `.getr(index)`
  - Built in function gets a value at a given index
- `.length()`
  - Gets length of list
- `.insert(value)`
  - Inserts value to the end of the list

### 3.13 functions

- Functions structure involves
  - Return type
  - Identifier
  - parameter list
  - Code block that is executed when function is called
  - Return statement in code block
  - Eg. `void functionID(){ //code block }`
  - If function type is void no return is required
- Functions are called by calling the function id (i.e typing function id followed by `()`)
  - `functionId()` ;

- Function return values can be stored in variables

### **3.14 Additional resources**

*For users that have an understanding of how bnf works you can view this link and get a deeper insight.*

<https://gitlab.com/computing.dcu.ie/lynchs43/2020-ca400-template-repo/blob/master/docs/documentation/BNFDocument.pdf>

## **4. Basic Program Tutorials**

4 basic programming tutorials have been developed for readers to get started with mona programming language and can be found at link:

[https://drive.google.com/drive/folders/1\\_VOyGigSFhIFGkUK5X4QSoaDvRaG8QD1?ths=true](https://drive.google.com/drive/folders/1_VOyGigSFhIFGkUK5X4QSoaDvRaG8QD1?ths=true)

Available Tutorials:

1. Hello world
2. Data-types
3. If while lists
4. odds even