## Riphah International University

Riphah International University

**BSCS\_8 (A&B \_Spring 2025)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course Title: | Compiler Construction | | | | Course Code: | | CS4703 | Credit Hours: | 3 |
| Instructor: | Ms. Sidra Rani | | | | Program Name: | | BSCS | | |
| Semester: | 8 | Batch: |  | Section: | 8A | | Date: | 13 March 2025 | |
| Due Date: | **20 March 2025** | | | | Maximum Marks: | | |  | |
| Student’s Name: | Mehboob Waqar  Muhammad Fahad | | | | SAP-ID | 37134  37125 | | | |

## Project Deliverable 01

Propose a new programming language with the following specifications:

* **Language Name**
* **Compiler Name**
* **Language Constructs** (including variables, data types, keywords, conditional statements, iterative statements, and functions)
* **Transition Diagrams**

**Proposal for a New Programming Language CodeForge**

### **1. Language Details**

* **Language Name:** CodeForge
* **Compiler Name:** CodeForgeCompiler

**2. Language Constructs**

**Variables**

| **Feature** | **Description** |
| --- | --- |
| **Declaration Symbol** | Variables are initialized using the keyword, the variable's name. |
| **Example** | Sring name = “Mehboob”;  nmb age = 12; |

**Data Types**

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | **Data Type** |  | | **Description** |
| nmb | Integer type for whole numbers |
| flat | Floating-point numbers |
| Sring | String type for text data |
| buul | |  |  | | --- | --- | |  | Boolean type (true or false) | |
| cter | Character type, enclosed in single quotes |

**Keywords**

|  |  |
| --- | --- |
| **Category** | **Keywords** |
| **Conditional Statements** | oye (for if condition),  naioye (alternative condition),  chaloye (for else-if condition) |
| **Loops** | first (for loop),  wish (while loop),  last wish (do-while loop),  brk (for break) |
| **Functions** | Ex (for function declaration) |
| **Return** | rat |
| **Input/Output** | enter , printO |
| **Switch** | option |
| **Cases** | pick |
| **Default** | autopick |

### **Loops**

#### **Keyword & Example Code**

|  |  |
| --- | --- |
|  |  |
| **first (for loop)** | first (nmb i = 1; i <= 3; i++) {  printO<<”Number”<<i;  } |
| **wish**  **(while loop)** | nmb count = 1;  wish (count < 3) {  printO<<count;  count++;  } |
| **Last wish**  **(do-while loop)** | nmb count = 1;  last {  printO<<count;  count++;  } wish (count > 4); |
| **brk** | first (nmb i = 1; i <=5; i++) {  oye (i == 5) {  brk;  }  printO<<i;  } |

### **Functions**

**Function Declaration and Example Code**

Ex my() {

printO<<”Mehboob Waqar”;

}

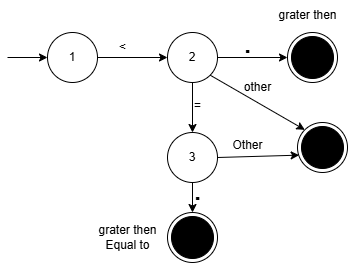
my();

**Operators:**

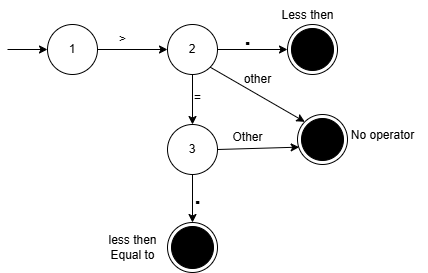
|  |  |  |
| --- | --- | --- |
| **Operator** | **Symbol** | **Description** |
| Addition | +. | Adds two values |
| Subtraction | -. | Subtracts two values |
| Multiplication | \*. | Multiplies two values |
| Division | /. | Divides two values |
| Modulus | %. | Finds the remainder |
| Assignment | =. | Assigns a value |
| Equal to | ==. | Checks if two values are equal |
| Not equal | !=. | Checks if two values are not equal |
| Greater than | >. | Checks if left is greater than right |
| Less than | <. | Checks if left is smaller than right |
| Greater or equal | >=. | Checks if left is greater than or equal to right |
| Less or equal | <=. | Checks if left is smaller or equal to right |
| Logical AND | &&. | Both conditions must be true |
| Logical OR | ||. |  |
| Logical NOT | !. | Reverses a condition |
| Increment | ++. | Increases a value by 1 |
| Decrement | --. | Decreases a value by 1 |

**3. Transition Diagrams:**

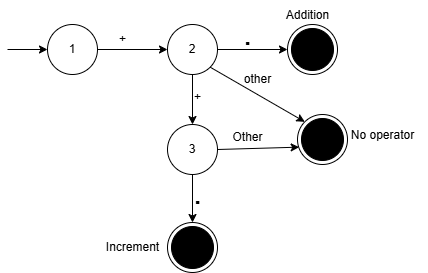
* **GraterThen or equal**



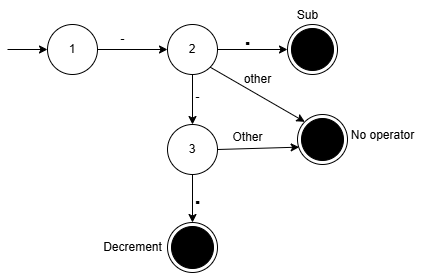
* **LessThen or equql**

****

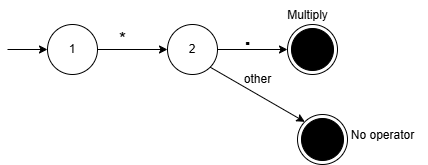
* **Addition & Increment**

****

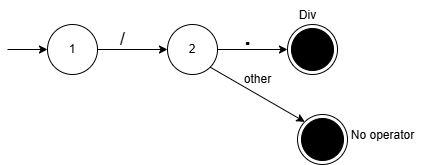
* **Subtraction & Decrement**

****

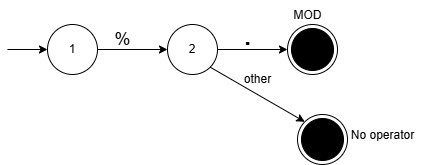
* Multiply



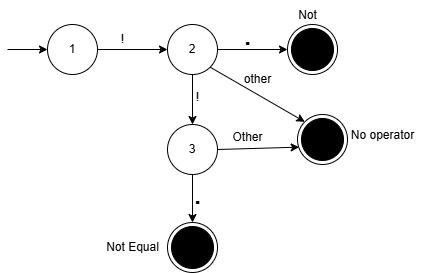
* Divide



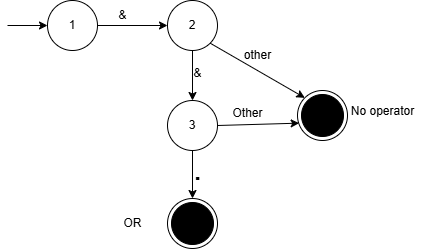
* MOD



* Not or Not equal



* And



* OR

