*Compiler Design*

***Data Types:***

* ***Integer:*** Whole numbers (1,2,4, …)
* ***Float:*** Decimal numbers (1.2, 5.4, ….)
* ***String:*** Text enclosed in double quotes (“Welcome”)
* ***List:*** A collection of elements enclosed in square brackets ([1, 2, 3])

***Variables:***

To assign a value to a variable, use the dollar sign $ followed by the variable name and the assignment operator ‘=’

Here's an example:

**$ variable = 10**

***Operators:***

* **Arithmetic Operators**: +, -, \*, /, ^ (power).
* **Comparison Operators**: ==, !=, >, <, >=, <=.
* **Logical Operators**: AND, OR, NOT.

***Control Structures:***

***If-Else:***

Use **IF**, **ELSEIF**, and **ELSE** for conditional expressions:

**IF** condition **THEN**

Code to execute if condition is true

**ELSEIF** another\_condition **THEN**

# Code to execute if another\_condition is true

**ELSE** Code to execute if no condition is true

***For Loop:***

The **FOR** loop iterates over a range of values:

**FOR** $ I = 0 **TO** 10 **STEP** 2 **THEN**

# Code to execute for each iteration

***While Loop:***

The **WHILE** loop continues executing as long as a condition is true:

**WHILE** condition **THEN** # Code to execute while condition is true

***Error Handling:***

It provides basic error handling through custom error types. Common errors include:

* ***Syntax Errors***: Errors caused by incorrect syntax.
* ***Runtime Errors****:* Errors occurring during script execution, such as division by zero.

Here's an example of a runtime error due to division by zero:

**$x = 10 / 0 # This will cause a runtime error**

***Code Examples:***

Below are some basic code examples:

***Arithmetic Operations:***

**$result = 5 + 3 \* 2 # Result is 11**

***Conditional Expressions:***

**IF $x == 10 THEN print("X is 10")**

***Loops:***

**FOR $ i = 0 TO 10 STEP 1 THEN print($i)**