# Nova Programming Language — User Manual

Welcome to the Nova programming language. Nova is a simple, educational programming language designed for students learning compiler construction.
It offers a clean syntax and minimal features to help you understand the fundamental phases of compilation, including lexical, syntax, and semantic analysis.

## 1. Getting Started

Nova programs are written in plain text files with the extension .nova. To run a Nova program, compile it using your Nova compiler or interpreter.

Example command (if using a compiler built with Flex and Bison):

```
> ./nova_compiler example.nova
```

## 2. Basic Syntax

Nova is case-sensitive and ignores whitespace except within strings. Each statement ends with a newline or semicolon.
Keywords are reserved and cannot be used as variable names.

Reserved Keywords:
var, int, float, str, input, output, if, else, end, while, for, do, true, false, and, or, not

## 3. Variables

Variables are declared with the var keyword followed by a type and an optional initializer. Supported types are int, float, and str.

Example:

```
var int age = 21
var float pi = 3.1416
var str name = "Alice"

# Assigning values later
age = age + 1
name = name + "Johnson"name = name + " Johnson"
```

## 4. Input and Output

Nova provides simple I/O functions using input and output.

- input var — Reads user input into a variable (type must match).
- output expr1, expr2, ... — Prints one or more expressions separated by spaces.

Example:

```
input name
input age
output "Hello,", name, "You are", age, "years old."
```

## 5. Comments

Comments allow you to document your code.

- Single-line comments start with // and go to the end of the line.
- Multi-line comments are enclosed between /* and */.

Example:

```
// This is a single-line comment
/*
This is a
multi-line comment
*/
```

## 6. Conditional Statements

Nova supports conditional branching using `when ... then ... else ... end`.
The `else` block is optional.

Example:

```
if age >= 18 then
   output "You are an adult."
else
   output "You are a minor."
end
```

## 7. Loops

Nova provides two loop types for iteration and condition-based repetition.

- Counting loop: for i = start to end do ... end
- Conditional loop: while (condition) do ... end

Example:

```
// Counting loop
for i = 1 to 3 do
    output "Hello!"

end
```

```
// Conditional loop
var int x = 5
while x > 0 do
   output "x =", x
   x = x - 1
end
```

## 8. Expressions

Expressions combine literals, variables, and operators to compute values.
Supported operators include:

- Arithmetic: +, -, *, /, %
- Relational: ==, !=, <, <=, >, >=
- Logical: and, or, not

Example:

```
var int a = 5
var float b = 2.0
output (a + 3) * b, a % 2 == 1 and not (b == 0.0)
```

## 9. Example Program

```
# Example: Counting Greetings
var str name = "User"
var int n = 3
output "Enter your name:"
input name
output "How many times to greet?"
input n
for i = 1 to n do
   if n > 5 then
      output "Hi", name, "(large count)"
   else
      output "Hi", name
   end
end
```

## 10. Conclusion

Nova is a lightweight programming language designed to help beginners understand the working principles of compilers.
Through simple syntax and structured constructs, Nova encourages learners to explore lexical analysis, parsing, and code generation.
This manual provides all necessary details to write and execute programs in Nova effectively.