

VeraZ Banglisch Programming Language

VeraZ — Code, manage, and print your results easily in Banglisch!

Welcome to VeraZ, a programming language based on Bangla that combines English and Bangla to make coding more approachable, particularly for people who are fluent in both languages. VeraZ is made to provide students with an entertaining and interesting way to learn compiler construction. You will be able to manage lists, automate processes, and carry out simple programming operations with this straightforward yet effective language.

With a language that feels natural to them, VeraZ is the perfect instrument for novices to grasp the fundamentals of lexical, syntactic, and semantic analysis.

1. Getting Started

To start using VeraZ, you need to have the VeraZ compiler or interpreter installed on your system. Once you have it set up, follow these steps to run your first program.

Example command (assuming you've built the compiler with Flex and Bison):

```
> ./veraz_compiler example.veraz
```

This command compiles your VeraZ program, which is written in .veraz files. Simply replace example.veraz with the name of the program you want to run.

2. Basic Syntax

case-sensitive: VeraZ is case-sensitive, meaning variable and Variable are treated as two different entities.

Whitespace: VeraZ ignores extra whitespace, except inside strings.

Statement Termination: Each statement is terminated either with a newline or a semicolon.

Here's an example:

```
banao List numbers = [1, 2, 3, 4]
```

Reserved Keywords:

```
banao, List, addKor, removeKor, sortKor, filterKor, showKor, if, else, end, for,  
pratek, true, false, loop, shart, jotoikkhon, chaluKor, kaj, poriborton
```

3. Variables

Defining Variables: Variables in VeraZ are defined using the banao keyword, followed by the type (e.g., List, int, text) and an optional initializer.

```
banao List friends = ["Riyad", "Sumi", "Tanveer"]
banao text username = "CoderBhai"
banao int maxScore = 100
```

You can also update the value of a variable:

```
friends = ["Shafiq", "Tariq", "Nabila"]
```

4. Input and Output

Reading Input: Use the read input command to get input from the user. The type of the input should match the variable type.

Printing Output: To print data to the screen, use the showKor command:

```
read username
read maxScore
showKor "Yo,", username, "your max score is", maxScore
```

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

VeraZ supports **if-else** conditions to control the flow of your program.

If-Else Statement

```
if maxScore >= 50 then
    showKor "Well done! You passed!"
else
    showKor "Try again next time."
end
```

7. Loops

Looping Through a List

You can use a loop to iterate through a List:

```
loop pratek member in friends do
    showKor member
end
```

While Loop

You can use a while loop to execute a block of code repeatedly as long as a shart (condition) is true.

```
banao int counter = 0
while counter < 5 do
    showKor "Counter:", counter
    counter = counter + 1
end
```

8. Expressions

8.1 Add an Item to a List

```
addKor fruits "orange"
```

8.2 Remove an Item from a List

```
removeKor fruits "banana"
```

8.3 Sort a List

```
sortKor fruits
```

8.4 Filter a List

```
filterKor fruits item == "apple"
```

9. Example Program

Example: Squad Management

This example demonstrates how to manage a list of friends, filter them based on conditions, and print the results.

```
banao List friends = ["Riyad", "Sumi", "Tanveer"]
banao text username = "CoderBhai"
showKor "Your squad is:"

loop pratek member in friends do
    showKor member
end

# Filter squad members who are active
filterKor friends member == "Riyad"
showKor "Active squad members:", friends
```

10. Conclusion

VeraZ is a programming language that uses a combination of Bangla and English to simplify programming ideas. It is intended to help students who are proficient in both languages understand and apply fundamental programming concepts in a more natural and interesting way. You should have no trouble creating your own VeraZ applications if you adhere to the concepts and examples given in this manual.

The syntax, variable declarations, control flow, list operations, and other fundamentals of the language were addressed in this handbook. Have fun with your coding!