iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: https://doi.org/10.22214/ijraset.2022.45210

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

Creating a General-Purpose Procedural Language for Programming in Kannada

Rajeshwari S[1], Sharath R[2], Priyanka M S[3], Priyadarshini S[4]
[1, 2, 3, 4]*Information Science and Engineering, East West Institute of Technology*

Abstract: ***To build a general-purpose procedural programming language in Kannada and the corresponding header file that understands the language and pre-processes the code typed in Kannada into C++ programming language which can then be compiled by a C++ compiler and produce corresponding output in Kannada. The entire compilation of the language will be built in a single and Integrated Development Environment (IDE) i.e. Microsoft Visual Studio which can be downloaded by anyone and start learning to code in Kannada by including the header file. Any programming logic can be typed entirely in Kannada, the syntax is created such that it is similar to C-syntax. Any person knowing Kannada and with little effort on understanding the language's syntax can start coding in Kannada without any prior knowledge in English or programming. This paper aims mainly at providing its users a programming experience in a regional language "Kannada". The users can edit, complier and thus observe the results also in Kannada. The user has to abide by a set of syntactic rules and thus program accordingly for better results.***

Index Terms: Kannada, header file, programming language, C++, pre-process.

I. INTRODUCTION

Knowledge of English language is a pre-requisite for computer programming (as of now) and programming is the most sought out skill which is being introduced in schools. Programming is a structured way to introduce students to problem solving and logical thinking. Additionally, it prepares them for a work market where artificial intelligence will rule. Lack of language proficiency makes it difficult to organize thoughts and effectively communicate them. A language in Kannada forces Karnataka students to think and code in their mother tongue, which is the best way to enhance logical thinking and creativity. This undermines confidence, which results in an entire generation that is poor in thought and expression with loss of confidence and self-respect. According to the report by Department of Education, Government of Karnataka. There are 60,817 Kannada medium schools in Karnataka. The percentage of students studying in class 8-10 in Kannada medium schools throughout Karnataka is 71.14% of all the students in class 8-10 in Karnataka (according to the Analytical report 2011-12 by the Department of education, Government of Karnataka).

Creating a programming language is to build a compiler that take the program typed and converts into machine instructions, to convert the program into machine instructions a compiler or an interpreter has to be built. To build a compiler they are 6 phases, only after all the 6 phases has be created the compiler can be used but the problem with building a compiler or a interpreter is that they are not modifiable i.e. any small change in syntax or semantics of the programming language then the entire 6 phases of the compiler has be rebuilt from the scratch and when create a new programming language in a non-English language there would be a lot of modifications since there isn't any standards existing.

The process of building a compiler through all its phases from scratch is very time consuming and is highly unmodifiable in future i.e. once the compiler is built any changes to the programming language cannot be made until all the 6 phases has been modified and reprogrammed. The code is typed in Kannada in any text editor by including a single header file and is compiled used the C++ compiler, the translation from the code typed in Kannada to C++ code happens the in pre-processor stage where the included header file is used to support the conversion from Kannada to C++ which is then converted to machine code using C++ compiler such as GNU compiler, intel C++ compiler, turbo C etc.

II. SYSTEM REQUIREMNETS

A. Software Requirements

1) Operating System: Any Operating System
2) Windows Terminal
3) Microsoft Visual Studio

B. Hardware Requirements

1) Key Board: Standard Keyboard (Kannada keyboard optional)
2) Monitor: Any color monitor
3) RAM : 2GB
4) Hard disk :500GB

III. SYSTEM DESIGN

The main aim of our project is to provide a platform for those who want to program using a language that is typed not in English but in Kannada. This would aid those who are not well versed in English, who know Kannada and want to learn programming.
The high level design uses 2 major components:

1) Preprocessor
2) Compiler

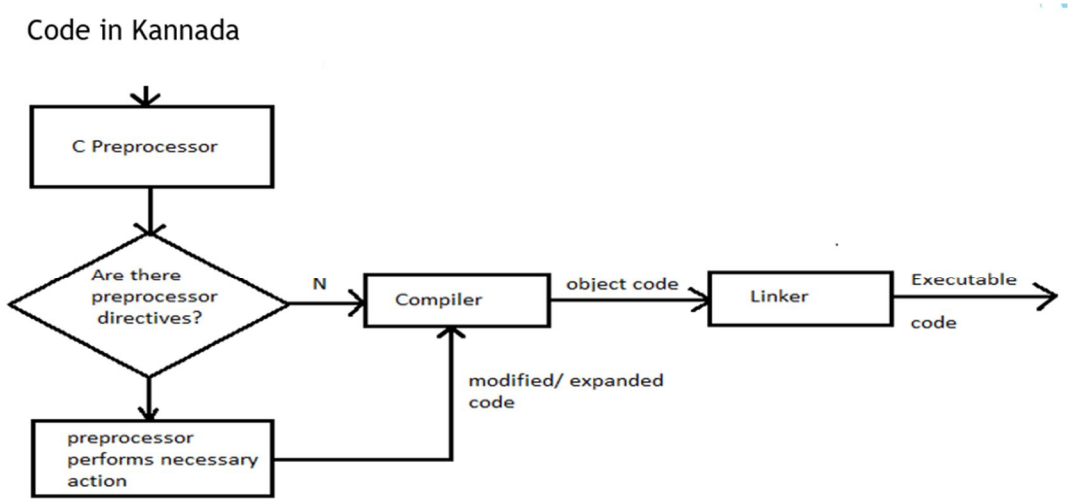


Fig 1. System Architecture

The program is typed in Kannada by including the created header file in any text editor supporting Unicode characters. The C++ preprocessor takes the code typed in Kannada and replaces the keyword in Kannada to the keywords understandable by the C++ compiler. After preprocessing the entire program is converted from Kannada to C++ language which can be compiled through any C++ compiler. There is a direct mapping between the datatypes in C++. Kannada programming is carried out using pre-processing capabilities of C/C++ compilers. A header file contains all the code necessary to facilitate Kannada programming. A header file is a file that contains shared macro definitions and C declarations for use by multiple source files. By adding a header file using the C preprocessing command "#include," the program can support coding in Kannada.

IV. IMPLEMENTATION

The header file <ಕನ್ನಡ> contains all the support needed for programming in Kannada. The keywords of the C++ Language is been #define into Kannada words, the program that is typed in Kannada will be using the standard namespace by default.

A. Datatypes

The datatypes that have been implemented in Kannada are:

1) ಅಂಕ*(int):* An integer type variable can store values that are zero, positive, and negative. The " keyword in the C language denotes the integer data type, which can be either signed or unsigned. If an integer variable's value is unsigned, it is by default regarded as positive.
2) ದಶಾಂಶ*(float):* The decimal values in this datatype can have up to 6 digits after the decimal point. The float data type has a storage size of 4 bytes, however like the "int" data type, this value may vary depending on the processor.

3) ಪದ(wstring) : This datatype is used store a character or a string of character in any language. The underlying datatype wstring is an Unicode datatype hence string of Kannada letters can also be stored. The default type of string that the C++ compiler detects is ASCII string, in order to specify the string as wide character string, the string has to be prefixed with a letter L. Every string that uses Kannada in the language now should be prefixed with an L.

Example : L "ಒಂದೂ ಪದ (ಸ್ಟ್ರಿಂಗ್)"

But using an English letter in a Kannada programming language is not desirable therefore the same functionality is implemented using the concatenation operator ## in the pre-processor directives. Hence now the Kannada character string can be specified using ಕ(). Example : ಕ(ಒಂದೂ ಪದ (ಸ್ಟ್ರಿಂಗ್))

B. Conditional Statements

The conditional statements that has been implemented in Kannada are :

ಆದರೆ (if) ಅಥವಾ (else): The ಆದರೆ statement controls conditional branching. The body of an ಆದರೆ statement is executed if the value of the expression is nonzero. The syntax for the ಆದರೆ statement has two forms.

Syntax :

selection-statement: ಆದರೆ (expression) statement

ಆದರೆ (expression) statement ಅಥವಾ statement

The test expression inside the parentheses is evaluated by the ಆದರೆ statement (). The statements inside the body of ಆದರೆ are carried out if the test expression is assessed as true. The statements inside the body of the ಆದರೆ are not performed if the test expression is evaluated to false. The statements inside the body of ಆದರೆ are carried out if the test expression is assessed as true. Statements contained in ಅಥವಾ body are not executed. Statements contained in the ಅಥವಾ body are carried out if the test expression is interpreted as false. Execution of statements included within the body of ಆದರೆ is skipped. The following is an examples of the ಆದರೆ ಅಥವಾ statement:

```
ಆದರೆ ( ಸಂಖ್ಯೆ > 0 )
   ಸಂಖ್ಯೆ++;
ಅಥವಾ
   ಸಂಖ್ಯೆ--;
```

In this example, the statement ಸಂಖ್ಯೆ++; is executed if ಸಂಖ್ಯೆ is greater than 0. if ಸಂಖ್ಯೆ is less than or equal to 0 then ಸಂಖ್ಯೆ--; statement is executed , Note that the statement forming the ಆದರೆ clause ends with a semicolon.

C. Looping Statements

ಗೆ(initialization ; condition; increment) : A ಗೆ() loop is a repetition control structure used to execute a single or a set of statements a specific number of times.

A Sample program to find the sum of n numbers using ಗೆ() loop:

```
#include <ಕನ್ನಡ>
ಅಂಕ ಮುಖ್ಯ() {ಬರೆ << ಕ("ನಿಮ್ಮ ಸಂಖ್ಯ ನೀಡಿ ") << ಅಂತ್ಯ;
        ಅಂಕ ಸಂಖ್ಯ, ಸಂಖ್ಯಿ;
        ಅಂಕ      ಮೊತ್ತ = ೦;
        ಓದು >> ಸಂಖ್ಯಿ;
        ಗೆ(ಸಂಖ್ಯ = ೦; ಸಂಖ್ಯ < ಸಂಖ್ಯಿ; ಸಂಖ್ಯ++) {
        ಮೊತ್ತ += ಸಂಖ್ಯ;
        }ಬರೆ << "1 + 2 + 3..... + " << ಸಂಖ್ಯ << " = " << ಮೊತ್ತ;}
```

ಆಗಿರುವವರೆಗೆ() : A ಆಗಿರುವವರೆಗೆ() loop is a type of repetition control structure that makes it simple to create loops that must run a certain number of times. In this case, a statement or group of statements may be either singular or collective. Any expression that meets the criterion is true, and a nonzero value does too. While the condition is true, the loop iterates. The line just after the loop receives program control when the condition is false.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

A Sample program to find the sum of n numbers using ಆಗಿರುವವರೆಗೆ() loop:

```
#include <ಕನ್ನಡ>
ಅಂಕ ಮುಖ್ಯ() {
        ಬರೆ << ಕ("ನಿಮ್ಮ ಸಂಖ್ಯ ನೀಡಿ ") << ಅಂತ್ಯ;
        ಅಂಕ ಸಂಖ್ಯ, ಸಂಖ್ಯ೨;
        ಅಂಕ      ಮೊತ್ತ = ೦;
        ಓದು >> ಸಂಖ್ಯ೨;
        ಸಂಖ್ಯ = ೦;
        ಆಗಿರುವವರೆಗೆ(ಸಂಖ್ಯ <= ಸಂಖ್ಯ೨)
        {ಮೊತ್ತ += ಸಂಖ್ಯ;
        ಸಂಖ್ಯ++;}
        ಬರೆ << "1 + 2 + 3..... + " << ಸಂಖ್ಯ << " = " << ಮೊತ್ತ;}
```

D. Operators

An operator is a symbol that instructs the compiler to carry out particular logical or mathematical operations. The following operators are available in Kannada:

1) Arithmetic Operators
2) Logical Operators
3) Assignment Operators
4) Relational Operators
5) Bitwise Operators

Since the underlying language is C++, every operator supported in C++ can be used in Kannada programming.

E. Example

A sample program in Kannada to print Fibonacci Series up to n digits.

```
#include <ಕನ್ನಡ>
ಅಂಕ ಮುಖ್ಯ()
 { ಅಂಕ ನ, ಸಂಖ್ಯ = 0, ಸಂಖ್ಯ೨ = 1, ನ೦ತರ = 0;

   ಬರೆ << ಕ("ಕಡೆಯ ಅಂಕೆಯನ್ನು ನೀಡಿ") << ಅಂತ್ಯ;
   ಓದು >> ನ;

   ಬರೆ << ಕ("ಫಿಬೊ್ಬನಕಿ ಸಂಕೆಗಳು: ") <<ಅಂತ್ಯ;
   ಗೆ (int ಶ = 1; ಶ <= ನ; ++ಶ) {
        ಆದರೆ (ಶ == 1) {
        ಬರೆ  << ಸಂಖ್ಯ << ಅಂತ್ಯ;
        ಮುಂದುವರಿಸು;
      }
      ಆದರೆ (ಶ == 2) {
        ಬರೆ << ಸಂಖ್ಯ೨ << ಅಂತ್ಯ;
        ಮುಂದುವರಿಸು;
      }
      ನ೦ತರ = ಸಂಖ್ಯ + ಸಂಖ್ಯ೨;
      ಸಂಖ್ಯ = ಸಂಖ್ಯ೨;
      ಸಂಖ್ಯ೨ = ನ೦ತರ;
```

ಬರೆ << ನಂತರ <<ಅಂತ್ಯ;}
ಮರುಳಿಸು 0}

V. RESULTS AND SNAPSHOTS

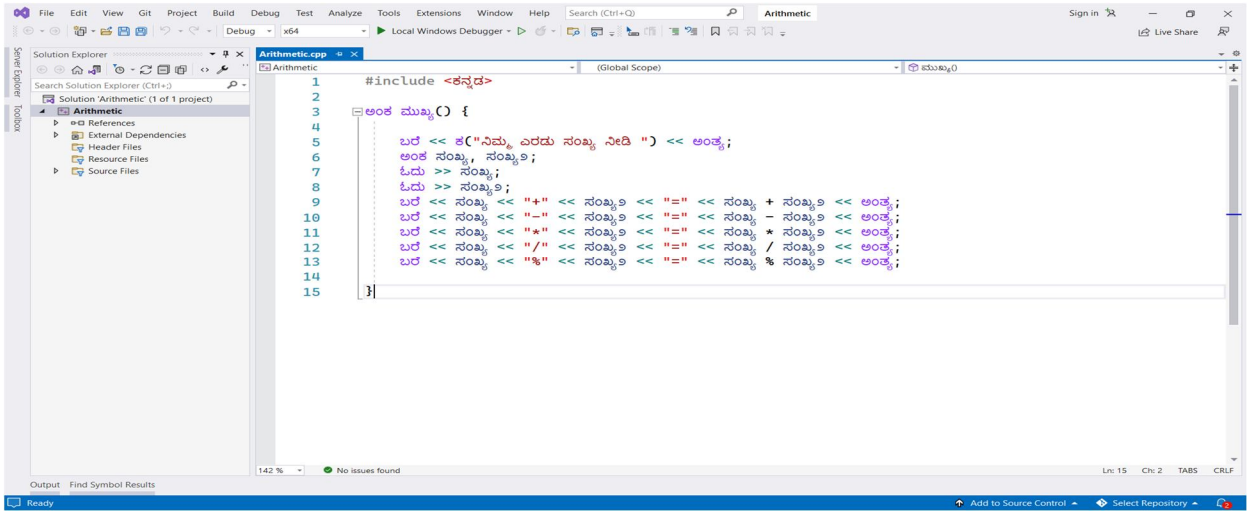


Fig.2. A sample program implementing arithmetic operations in Kannada.

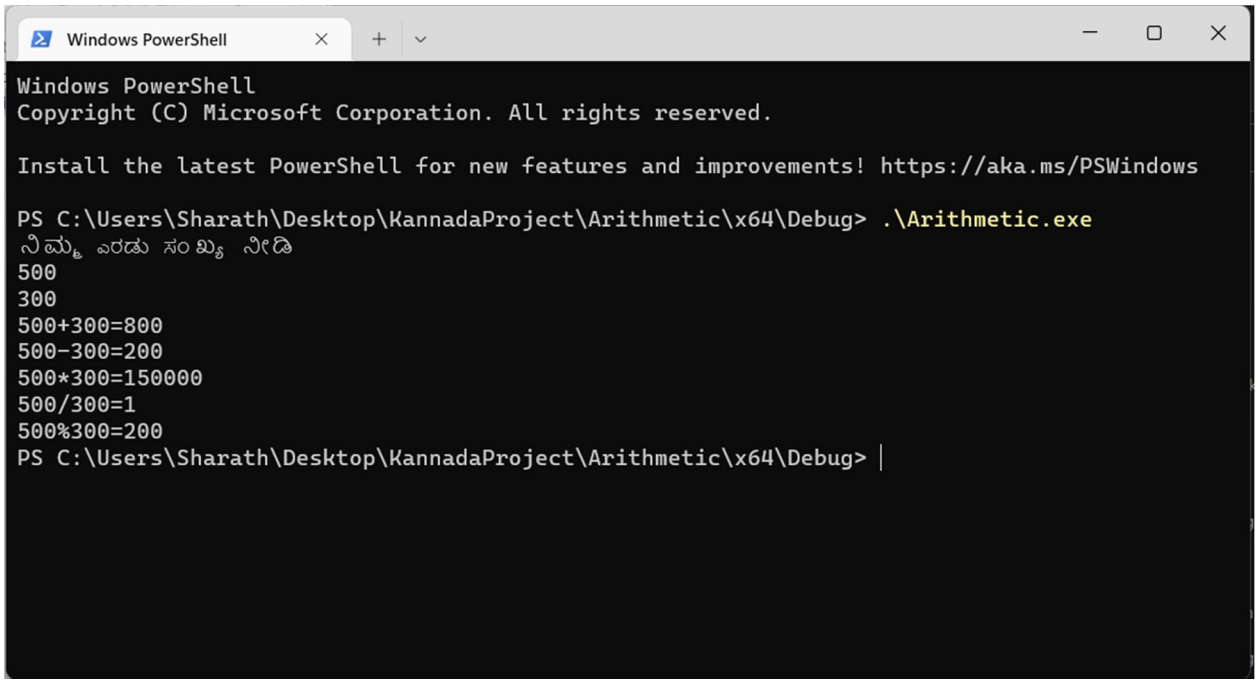


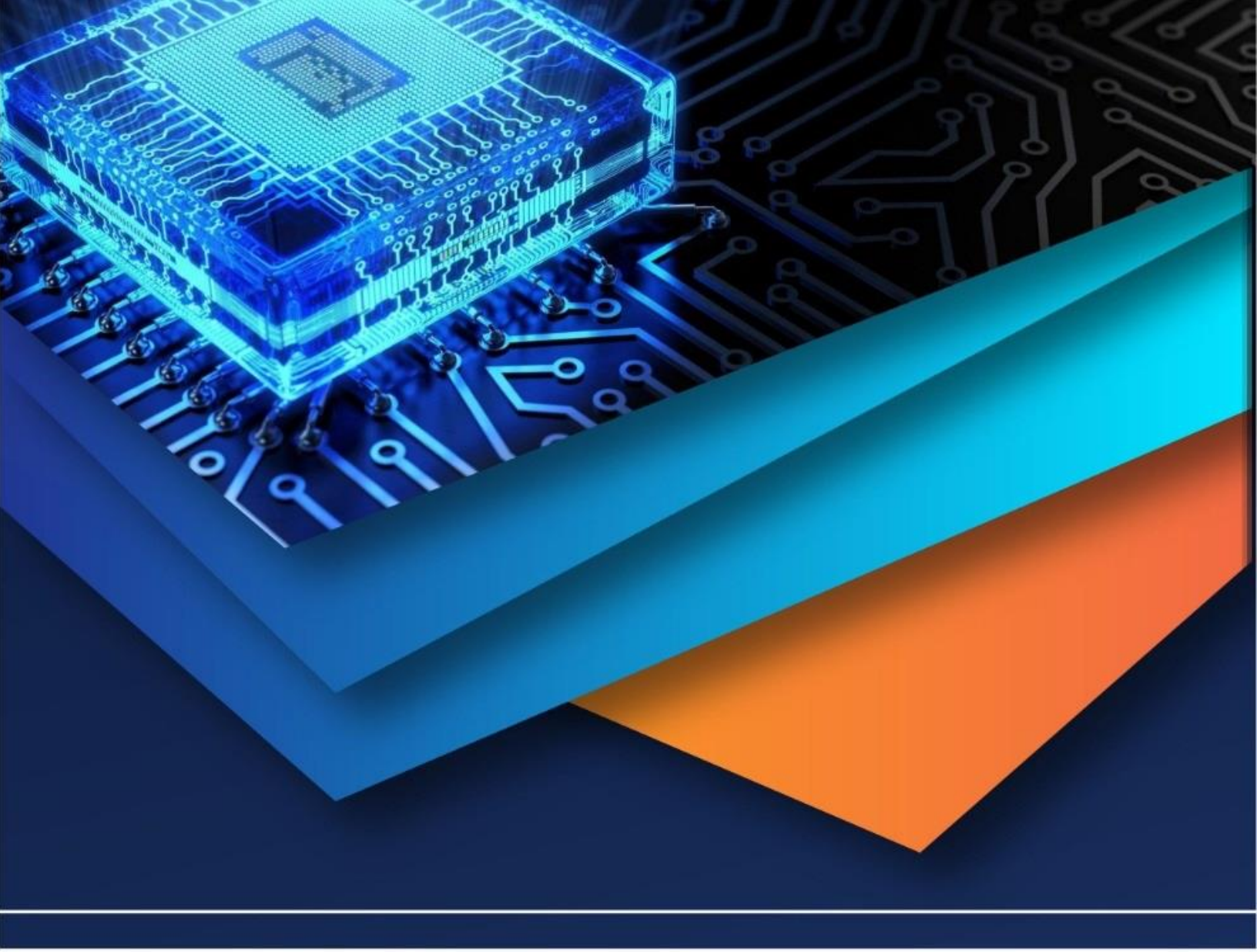
Fig.3. Output in Windows Terminal of the above program.

VI. CONCLUSION

Using the proposed system, any person knowing Kannada and with little effort on understanding the language's syntax can start coding in Kannada without any prior knowledge in English or programming. The Header file, Microsoft Visual Studio and Windows Terminal can be downloaded and installed in any computer, This project can be used to learn programming by anyone without any prior knowledge in English. The proposed system can be implemented in Kannada Medium Schools in Karnataka which are 74 percent of total schools in Karnataka.

REFERENCES

[1] PraNaMa – scripting language in Kannada - International Journal of Computer Applications (0975 – 8887) Volume 58– No.20, November 2012
[2] The Principles of Compiler Design, 4th edition by Jeffery Ulmann et.al.
[3] The C++ programming language, Bjarne Stroustrup
[4] http://en.wikipedia.org/wiki/compilers
[5] http://www.google.com/transliterate/Kannada -Transliteration APIs
[6] https://en.wikipedia.org/wiki/GNU_Compiler_Collection



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY