PROJECT REPORT

COURSE TITLE : Compiler Lab
COURSE CODE : CSE 352

SUBMITTED TO :

Md. Tasnin Tanvir
Lecturer,
Department of Computer Science and Engineering
University of Information Technology & Sciences
(UITS)

SUBMITTED BY :

NAME : Tonny Talukder
ID Number : 0432220005101050
Section : 6A1
Batch : 52
Department : CSE

DATE OF SUBMISSION : 26/05/2025

TONNYLang – A Custom Programming Language

1. Objective

To create a fictional programming language called TONNYLang **that uses Bengali-style names for common programming functions. This is intended to help beginners relate better with programming logic in their own language.**

2. Introduction

TONNYLang is a beginner-friendly language wrapper built on C++ syntax but with function and control structure names adapted to Bengali words. It provides a cultural and linguistic bridge for new learners to understand programming concepts more easily.

3. Language Design

A total of 25 essential programming concepts were translated into Bengali-style keywords. Here are some examples:

Serial	Original Function	TONNYLang Name
1	sum	Jogkoro
2	sub	komao
3	mul	Gunkoro
4	div	vagkoro
5	print	dekhao
6	input	nau
7	if	jodi
8	else	nahole
9	for	protibar
10	while	jotokkhon
11	switch	poriborton
12	case	obostha

13	break	thamao
14	continue	cholteythako
15	function	kajkoro
16	return	ferotkoro
17	import	niyeaso
18	class	dhoron
19	try	cheshtakoro
20	catch	dhoro
21	throw	churedao
22	true	shotti
23	false	mittha
24	and	ebong
25	or	othoba

4. Tools Used

- Language: C++
- IDE: Code::Blocks
- File Structure:
  - `main.cpp` – Main file using TONNYLang functions
  - `new.h` – Header file with all custom-defined functions

5. Implementation

File: `main.cpp`
This file takes two inputs from the user and performs:

- Addition (`Jogkoro`)
- Subtraction (`komao`)
- Multiplication (`Gunkoro`)
- Division (`vagkoro`)

Also demonstrates:

- `if-else` (jodi-nahole)
- `for`, `while` loops (protibar, jotokkhon)
- `switch-case` (poriborton, obostha)
- `class` (dhoron)
- `try-catch-throw` (cheshtakoro, dhoro, churedao)

CODE :

1. main.cpp

```
#include<bits/stdc++.h>
#include "new.h"
using namespace std;

int main() {
   dekhao("TONNYLang e tomar shagotom!");

   dekhao("Doya kore duita songkha dao:");
   int a = nau();
   int b = nau();

   int jog = Jogkoro(a, b);
   int biyog = komao(a, b);
   int gun = Gunkoro(a, b);
   int vag = vagkoro(a, b);

   dekhao("Jogfol: " + to_string(jog));
   dekhao("Biyogfol: " + to_string(biyog));
```

```
dekhao("Gunfol: " + to_string(gun));
dekhao("Vagfol: " + to_string(vag));

if (a > b) {
   dekhao("a b theke boro");
} else {
   dekhao("nahole b boro");
}

dekhao("Protibar loop:");
for (int i = 1; i <= 3; i++) {
   cout << "protibar: " << i << endl;
}

int c = 0;
dekhao("Jotokkhon loop:");
while (c < 3) {
   cout << "jotokkhon: " << c << endl;
   c++;
}

int d = 2;
switch (d) {
   case 1:
      dekhao("obostha 1");
      break;
   case 2:
      dekhao("obostha 2");
      break;
   default:
      dekhao("konobosthai nai");
}

for (int i = 0; i < 5; i++) {
   if (i == 2) continue;
   cout << "cholteythako: " << i << endl;
}
```

```
    int result = kajkoro(4);
    dekhao("kajkoro result: " + to_string(result));
    cout << "ferotkoro: " << ferotkoro(result) << endl;

    dhoronManush amr;
    amr.nam = "Tonny";
    amr.boyosh = 21;
    amr.infoDekhao();

    cheshtakoro_dhoro_example();

    bool shotti = true;
    bool mittha = false;

    if (shotti && !mittha) {
        dekhao("Eta shotti ebong mittha noy.");
    }

    if (shotti || mittha) {
        dekhao("Eta shotti othoba mittha.");
    }

    return 0;
}
```

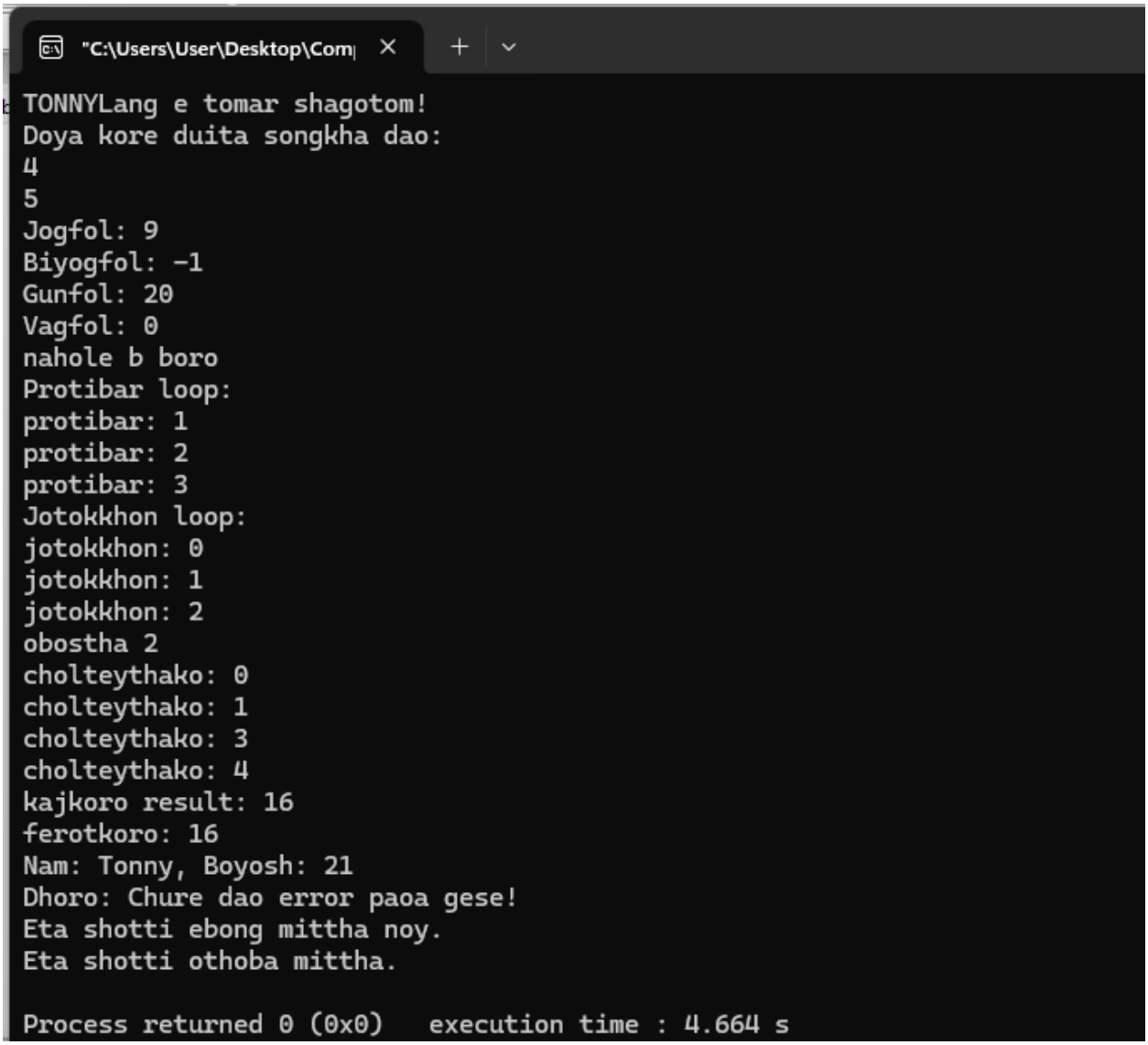
2. new.h

```
#ifndef NEW_H
#define NEW_H

#include<bits/stdc++.h>
using namespace std;
```

```
// 1. sum -> Jogkoro
int Jogkoro(int a, int b) {
   return a + b;
}

// 2. sub -> komao
int komao(int a, int b) {
   return a - b;
}

// 3. mul -> Gunkoro
int Gunkoro(int a, int b) {
   return a * b;
}

// 4. div -> vagkoro
int vagkoro(int a, int b) {
   if (b != 0) return a / b;
   else {
      cout << "0 diye vag kora jabena!" << endl;
      return 0;
   }
}

// 5. print -> dekhao
void dekhao(string msg) {
   cout << msg << endl;
}

// 6. input -> nau
int nau() {
   int x;
   cin >> x;
   return x;
}

// 15. function -> kajkoro
```

```
int kajkoro(int x) {
   return x * x;
}

// 16. return -> ferotkoro
int ferotkoro(int x) {
   return x;
}

// 19, 20, 21: try-catch-throw
void cheshtakoro_dhoro_example() {
   try {
      throw runtime_error("Chure dao error paoa gese!");
   }
   catch (exception &e) {
      cout << "Dhoro: " << e.what() << endl;
   }
}

// 18. class -> dhoron
class dhoronManush {
public:
   string nam;
   int boyosh;
   void infoDekhao() {
      cout << "Nam: " << nam << ", Boyosh: " << boyosh << endl;
   }
};

#endif
```

OUTPUT :



7. Conclusion

TONNYLang successfully demonstrates how programming fundamentals can be localized for better understanding. This fictional language builds a meaningful bridge between logic and native language for first-time programmers. It can be extended further to include arrays, recursion, file handling, etc.