

Compiler Design Life Cycle And Comparison of Programming Languages

¹Ali A. AL-Bakhrani, ²Abdulnaser A. Hagar, ³Ahmed A. Hamoud, ⁴Bharti W. Gawali, ⁵Ramesh Manza

***Abstract**---Python is a popular object-oriented language for programming for Python. Sometimes programmers spend a lot of time on running the program, while at the same time on code size. It creates the code more complex and untrustworthy, that reduces code efficiency So many compilers are currently available, like C, Java, C++, C #, python and so on. The code optimizing techniques for a python compiler have been researched separately and we come across new strategies for code optimization which is a clever way to do python code. In this paper we have done a compiler application and it has been written in five programming languages which we mentioned above where we have made a comparison between those languages in order to reach the best suitable language for building the compiler that help the programmer and facilitate the construction process and also the time to execution code is less and also less memory and less code size.*

Keywords: Python, Code, Design, Programming Languages.

I. INTRODUCTION

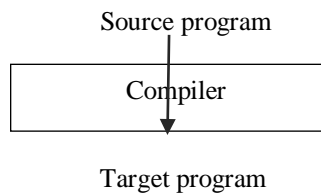
While we seek to study compilers, we must make a decision in choosing the language in which we will work on the application and that helps us in create compiler. With the multiplicity of programming languages, we decided in this paper to make a comparison between the programming languages(C, C ++, java, C #, Python) [1]. A compiler was built using all of these languages so that this program performs the same work on all languages and it is reading the code from a text file and entering it in the process of handling errors to correct this compiler. Where this work is easy for the programmer to choose the appropriate programming language, which we focused in this comparison on three main things are the size of the code and the degree of complexity and time of execution of the code. In compiler design the process is to improve a piece of code (un-optimized code) without modifying its efficiency [2]. The optimized programmer is easily defined as a small program which also utilizes less memory and needs less runtime. The programmer will make basic mistakes that are harmless for small programs mostly when writing a program language. But the efficiency of any program is slowing as the program grows. An important variable is then to improve the quality of an optimization application code [3]. Using different optimization

1,2,4,5 Department of Computer Science & Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad- 431004, India 3Department of Mathematics, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad - India.

methods, codes can be optimized with the purpose of high efficiency without impacting the original (existing) algorithm and final output. While calculating results, you must choose an algorithm that performs rapidly and the computer available resources are used. In theory, optimization of code needs less storage and more efficient implementation of program rules and algorithms for the program section, with a view to creating code more efficient. Optimization is defined as scalability of higher standards and scalability of lower levels [4]. The programmers who actually afford abstract entities also take account the broader framework of the task of automating the design of a system are usually able to optimize their optimisation. On the other hand, low-level optimization is made on the basis of source code in computer instructions [5].

About compiler

A compiler is a program that can read a program in one language-the *source* language-and translate it into an equivalent program in another language-the *target* language; see Fig. 1.1. An important role of the compiler is to report any errors in the source program that it detects during the translation process. A compiler translates the code written in one language to some other language without changing the meaning of the program. Computers are a balanced mix of software and hardware. Hardware is just a piece of mechanical device and its functions are being controlled by a compatible software.



An *interpreter* is another common kind of language processor. Instead of producing a target program as a translation, an interpreter appears to directly execute the operations specified in the source program on inputs supplied by the user, as shown in Fig. 1.2

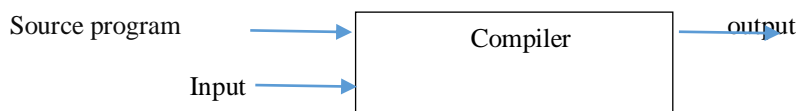


Figure 1.2: An interpreter

II. METHODOLOGY

Since the methodology is considered the auxiliary means used to complete any research. We have used in this paper several tools that helped us in the completion of the work and that have an effective role in giving us accurate results, and to you explain these tools in detail step by step:

II.1 J

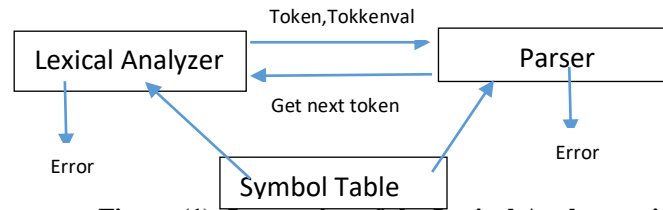


Figure (1): Interaction of the Lexical Analyzer with the Parse

Table (1) Symbol Table

Keyword:
(one of)

abstract	continue	for	new	switch
assert	default	if	package	synchronized
boolean	do	goto	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

II.2 Create a file to write code (Program): We have used a text file in which the code to be check is written as the code and the programming sentences in any language are written and examined and compared. Is it written the first true mistake depending on the symbol table [7].

II.3 Read a file (program/code): We have used the read function for reading the file, which reads the file line by line. It can be illustrated with the following example :
String arrofprog[]=br.read());

II.4 Extract tokens from file using function called spilt():This function is the most important teaching in the program, as it works on dividing the code to be examined into parts in order to facilitate the process of checking the code where the division is. Depending on the value sent in a parameter, we depend on the distance between each word where it works on dividing. Store the result in a single matrix. String arrofprog[]=br.read().spilt(“ “);

II.5 Match tokens : In this process, we used the forloop () sentence because it works to pass over all the elements by doing a full cycle of all the elements and comparing them with the key word file, for comparison using the IF() statements that support the client of comparison and matching between the code file and the comparison file.

II.6 Generate the output (error or correct): We have adopted a process that evaluates the comparison, or in other words, the result of the comparison with two main values: correct or

error, which informs us by giving us the word that was checked, followed by its true value, is it correct or error?

III. DISCUSSION

III.1 C Language Introduction

C is a language of process programming. It was first developed in 1972 by Dennis Ritchie. It was primarily designed to write an operating system as a machine programming language. These features make C language ideal for system programs like the installation of an operating system or the compiler creation. These provide low-level access to memory, a basic collection of keypads and clean style. C is what is known as a language compiled, this means you have to run the software through a C compiler when you write your C program to transform your software into an executable that the machine will run. The C program is the readable type for human beings, while the binary from the compiler is the readable and functional form for computers.

Table (2) Structure of C program

Structure of C program	
Header	#include<stdio.h>
Main()	int main() {
Variable declaration	int a=10;
Body	print(“%d”,a)
Return	return 0; }

Example compiler using C:

```
// C program to illustrate strings
```

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <assert.h>

#include <time.h>

char** string_split(char* a_str, const char a_delim)
{
    char** Result = 0;
    size_t count = 0;
    char* temp = a_str;
    char* last_space = 0;
    char delim[2];
    delim[0] = a_delim;
    delim[1] = 0;
    /* Count how many elements will be extracted. */
    while (*temp)
    {
        if (a_delim == *temp)
        {
            count++;
            last_space = temp;
        }
        temp++;
    }
    /* Add space for trailing token. */
    count += last_space < (a_str + strlen(a_str) - 1);
    /* Add space for terminating null string so caller
    knows where the list of returned strings ends. */
    count++;
    Result = malloc(sizeof(char*) * count);
    if (Result)
    {
```

```
size_t idx = 0;

char* tok = strtok(a_str, delim);

while (tok)
{
assert(idx < count);

*(Result + idx++) = strdup(tok);

token = strtok(0, delim);

}

assert(idx == count - 1);

*(Result + idx) = 0;

}

return Result;

}

int main()
{
clock_t startTime, endTime;

startTime = clock();

char keyword[] = "for ( int ) if else while do switch ";

char prog[] = "for ( int i = 0 ; i < 5 ; i++ )";

char** tok;

printf("Programe=[%s]\n\n", prog);

tok = string_split(prog, ' ');

if (tok)
{
int i;

for (i = 0; *(tokens + i); i++)
{

printf(" %s is Correct \n", *(tokens + i));

free(*(tok + i));

}

printf("\n");

}
```

```
free(tok);  
}  
endTime = clock() ;  
eleipsTime = endTime - startTime;  
double time_excution = ((double)eleipsTime)/CLOCKS_PER_SEC;  
  
printf("The totale time is %f Seconds To Execute \n", time_excution);  
return 0;  
}
```

Output the program C

```
Programme=[for ( int i = 0 ; i < 5 ; i++ )]  
  
for is Correct  
( is Correct  
int is Correct  
i is Correct  
= is Correct  
0 is Correct  
; is Correct  
i is Correct  
< is Correct  
5 is Correct  
; is Correct  
i++ is Correct  
) is Correct  
  
The totale time is 0.076 milliseconds To Execution
```

III.2 C++ Programming Language

C++ has been created by Bjarne Stroustrup as a generalized object-oriented (OOP) programming language and is also an extension of C language. C++ can be coded in "C" format or "object-oriented." It can be programmed as well in certain instances and is an efficient example of hybrid language. C++ is regarded to be an intermediate language as both high and low language features are encapsulated. Originally, the language was related to as "C with classes" due to its high many C language characteristics and an extra class element, but in 1983 it was called C++ [2]. C++ is

one of the most popular languages primarily utilized with system/application software, drivers, client-server applications and embedded firmware. The main highlight of C++ is a collection of predefined classes, which are data types that can be instantiated multiple times. The language also facilitates declaration of user-defined classes. Classes can further accommodate member functions to implement specific functionality. C++ Compiler: That you will need a C++ compiler to debug this file once you have a text editor and save the program in a file with '.CPP' extension. A compiler is a computer program that transfers high-level language to a low-level machine. In other words, it is translated into any other computer language that your computer understands writing in a programming language. In exchange to compile a C++ we need a compiler to transform the C++ source code into machine codes. Particulars about compiling on multiple platforms are provided below.

Table (3) Structure of C++ program

Structure of C++ program	
Header	#include<iostream.h>
Main()	int main() {
Variable declaration	int a=10;
Body	Cout<<a;
Return	return 0; }

Example compiler using C++

```
#include <bits/stdc++.h>
#include <boost/algorithm/string.hpp>
#include <time.h>
using namespace std;
int main()
{
clock_t startTime;
```



```
startTime = clock();  
  
string programeText("for ( int i = 0 ; i < 5 ; i++ )");  
  
string keywords("for if else ifelse while do switch default break ; 5 < ( ) = i i++");  
  
vector<string> prog,keyword;  
  
boost::split(prog, programeText, boost::is_any_of(" "));  
  
boost::split(keyword, keywords, boost::is_any_of(" "));  
  
int flag;  
  
for (int i = 0; i < prog.size(); i++)  
{  
for (int j = 0; j < keyword.size(); j++)  
{  
if(prog[i]==keyword[j])  
{  
flag=1;  
break;  
}  
else{  
flag=0;  
}  
}  
if(flag==1){  
cout<<prog[i]+" Correct"<<endl;  
}else  
{  
cout<<prog[i]+" Error"<<endl;  
}  
}  
  
endTime= clock() ;  
  
elipesTime= endTime- startTime  
  
double time_excution = ((double)elipesTime)/CLOCKS_PER_SEC; // in seconds  
cout<<" seconds to execute is " << time_excution<<endl;  
  
return 0;
```

```
}
```

Output the program C++

```
for Correct  
( Correct  
int Error  
i Correct  
= Correct  
0 Error  
; Correct  
i Correct  
< Correct  
5 Correct  
; Correct  
i++ Correct  
) Correct  
The totale time is 0.135 milliseconds To Execution
```

III.3 C # Programming Language

C #is an easier-to-use, type-safe object-oriented language that helps programmers to develop a variety of safe and robust. You can use C# to create Windows client programs, web XML systems, components distributed, server user applications, database applications, and much more. Visual C# delivers advanced code writing, simple user interface design, constructed-in debugger and many other equipment, making it easier to develop C# language and .NET Framework-based applications. C# programming language is – a modern high level language. When a programmer uses C#, he gives commands in high level, like from the position of a general executive in a factory. The instructions given in the form of programs written in C# can access and control almost all computer resources directly or via the operating system. Before we learn how to write simple C# programs, let's take a good look at the different stages of software development, because programming, despite being the most important stage, is not the only one [1].

Table (4) structure C# language

Structure of C# program	
Header	Using System;

Class	Class Compiler{
Main()	static int Main(string[] args) {
Variable declaration	int a=10;
Body	Console.WriteLine(a);
Return	return 0; } }

Example compiler using C#

```
using System;
class Compiler {
static void Main(string[] args)

{System.Diagnostics.Stopwatch sw = System.Diagnostics.Stopwatch.StartNew();
String progtxt = "forr ( int = 0 ; i < 5 ; i++ )";
String keyword = "for int if else ifelse while do switch default break ; 5 < ( ) = i i++";
char[] spearator = { ' ' };
int flage=0;
String[] proglis = progtxt.Split(spearator, StringSplitOptions.None);
String[] keywordlist=keyword.Split(spearator, StringSplitOptions.None);
foreach(String s in proglis)
{      foreach(String a in keywordlist){
if(s ==a ){
flage=1;
break;
}else{
flage=0;
}      }
}
```

```
if(flage==1){  
    Console.WriteLine(s+" Correct");  
}else{Console.WriteLine(s+" Error");  
}  
}  
}  
sw.Stop();  
Console.WriteLine("The totale time is "+sw.ElapsedMilliseconds+"Milliseconds to excoutin");  
}  
}
```

Output the program C#

```
for Correct  
( Correct  
int Correct  
= Correct  
0 Error  
; Correct  
I Correct  
< Correct  
5 Correct  
; Correct  
i++ Correct  
) Correct  
The totale time is 7 Milliseconds to excoutin
```

III.4 E) Java Programming Language

Java is a programming language developed by **James Gosling** with other team members named **Mike Sheridan** and **Patrick Naughton** also called as **Green Team** in **1995** for **Sun** Microsystems for digital devices such as set-top boxes, televisions etc. Now, let us see in detail what is Java. Java is a widely-used language of programming specifically designed for the distributed web environment. It's the most popular Smartphone programming language, and it's also among the most popular for developing trimming-edge smartphones and the Internet of Things. Java has also been designed to appear like C++ but is simpler to get and implements a model for object-oriented programming. It is possible to use Java for developing full apps that can run or be distributed between servers and clients through a network. It can also be used as part of the website for the development of a tiny application component or applet. The program is robust. Contrary to C++ programs, Java objects contain no references to external data or other known objects. It ensures that

the address of the data stored in some other application or the code itself can not be included into the instruction, since either would terminate or crash the program or perhaps the operating system. In order to make sure consistency, JVM performs a number of checks on each entity. JRE refers to a runtime environment in which Java bytecode can be executed. It implements the JVM (Java Virtual Machine) and provides all the class libraries and other support files that JVM uses at runtime. So JRE is a software package that contains what is required to run a Java program. Basically, it's an implementation of the JVM which physically exists [8].

) **Table**

Structure of java program	
Header	package compiler; import java.io.*;
Class	public class Compiler {
Main()	public static int main(String[] args) {
Variable declaration	int a=10;
Body	System.out.println(a);
Return	return 0; } }

(Structure java language)

Example compiler using java

```
package compiler;

import java.io.*;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.logging.Level;
```

```
import java.util.logging.Logger;

/** @author Saja */

public class Compiler {

    public static void main(String[] args) throws FileNotFoundException, IOException {

        long startTime = System.currentTimeMillis();

        File fprog = new File("C:\\Users\\Saja\\Documents\\NetBeansProjects\\Compiler\\src\\compiler\\prog.txt");
        File fkeyword = new File("C:\\Users\\Saja\\Documents\\NetBeansProjects\\Compiler\\src\\compiler\\keyword.txt");
        BufferedReader br = new BufferedReader(new FileReader(fprog));

        BufferedReader br2 = new BufferedReader(new FileReader(fprog));

        String stkeyword = br2.readLine();

        String st;

        int flage = 0;

        while ((st = br.readLine()) != null) {

            String[] arrofstr = st.split(" ");

            String[] arrofkeyword = stkeyword.split(" ");

            for (String a : arrofstr) {

                for (String s : arrofkeyword) {

                    if (a.equals(s)) {

                        flage = 1;

                        break;

                    } else {

                        flage = 0;

                    }

                }

            }

            if (flage == 1) {

                System.out.println(a + " Correct");

            }

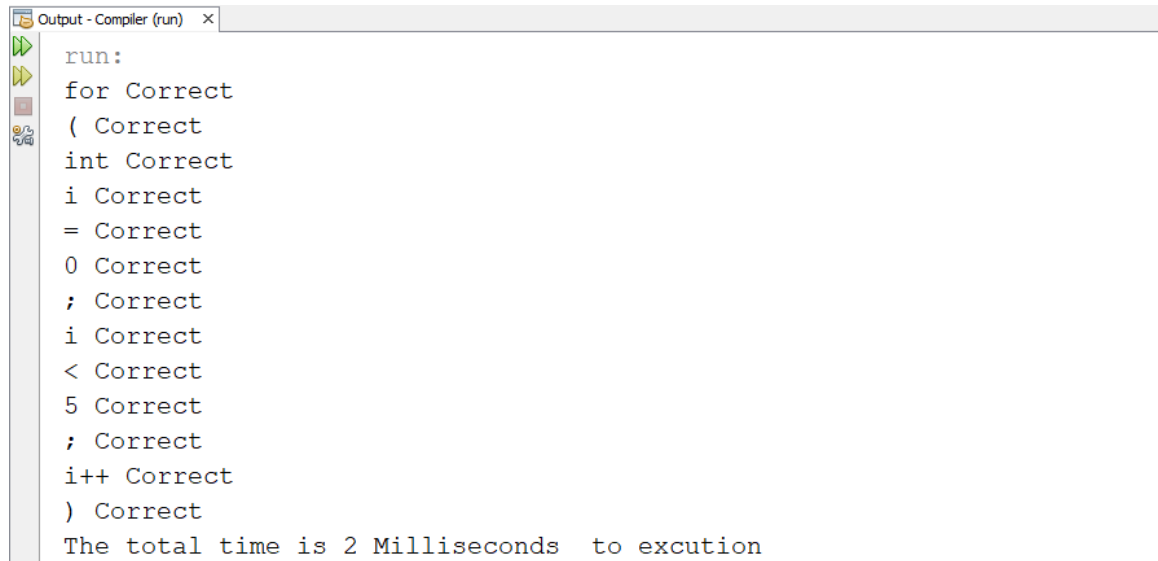
        }

    }

}
```

```
    } else {  
  
        System.out.println(a + " Error");  
  
    }  
  
    }  
  
    }  
  
    long stopTime = System.currentTimeMillis();  
  
    long elapsedTime = stopTime – startTime;  
  
    System.out.println("The total time is " + elapsedTime + " to execution");  
  
    }  
  
    }
```

Output the program java



```
Output - Compiler (run) x  
run:  
for Correct  
( Correct  
int Correct  
i Correct  
= Correct  
0 Correct  
; Correct  
i Correct  
< Correct  
5 Correct  
; Correct  
i++ Correct  
) Correct  
The total time is 2 Milliseconds to excution
```

III.5 Python Programming Language

Python is a strongly structured, object-oriented and dynamic semantic programming language. It is very marketable for fast development as well as for the use of existing components as scripting or glue language with a higher level of data structure, in combined effect with dynamic and dynamic binding. The simple, quick-to-learn syntax of Python emphasizes readability and further reduces program cost of maintenance. Python supports modules and packages that support the modifiability of programs and reuse of code [6]. The Python interpreter and the robust

standard library can be found on all major platforms in source and binary form, no expense and can be freely distributed for all major platforms. The higher productivity it provides often makes programmers love Python. The edit-test-debug process is incredibly fast, because there was no compilation step. It is simple to test Python programs: a bug or bad input will never cause a segmentation mistake. Rather, it generates an exception because the translator discovers an error. The interpreter prints a stack file when the application does not catch the exception. A source compiler allows local and global variables to be evaluated, random expressions evaluated, break points created, code line by line, etc. The debugger is published in Python, that shows the introspective power of Python.

Table (6) Structure python language

Structure of python program	
Header	Import time
Variable declaration	a=10
Body	print(a)

Example compiler using python

```
import time
start=time.time()
f=open(“prog.txt”)
f2=open(“keyword.txt”)
progtxt=f.read().split(“ “)
keyword=f2.read().split(“ “)
for a in progtxt:
    if a in keyword:
        print(a+“ Correct”)
    else:
        print(a+“ Error”)
end=time.time()
```


ellips=end-start

```
print("The totale time is ",ellips," Milliseconds to excoutin")
```

Output the program python



```
Run: Compiler x
C:\Users\Saja\PycharmProjects\ImageProcessLabe\venv\Scripts\python.exe C:/Users/Saja/
for Error
( Correct
int Correct
i Error
= Correct
0 Correct
; Correct
i Error
< Correct
5 Error
; Correct
i++ Correct
)
Error
The totale time is 0.0009968280792236328 Milliseconds to excoutin

Process finished with exit code 0
```

IV. CONCLUSION

In this paper and through the comparison we have concluded that there is a disparity in the construction of translators using programming languages where we worked to apply a translator program that works with the same work and with the same sentences that will be examined, it was discovered that there are criteria that distinguish one language from the other, for example:

- C language and Java language, there are two axes that we focus on, namely the degree of complexity of the code and the time of implementation. As for C language, the degree of complexity of the code is high and somewhat difficult and the code is long compared to Java, the code is small and somewhat uncomplicated implementation, we noticed that the reporting time was 0.076 milliseconds, and the implementation time in the Java language is 2 milliseconds, meaning if we calculated the ratio and proportion ($2 / 0.076 = 26.315$), while we implemented one program with Java at the same time, the C language would have implemented it 26 times, and this increases the speed the language .

-Python language comparison with the language of C# we found there is a difference where the Python language was distinguished by the ease of the code and the size of the code is very small also the implementation time took 0.000997 milliseconds while the C# language the degree of complexity of the code is average and the time of implementation is large it took time 7 milliseconds which means if We took the percentage ($7 / 0.000997 = 7021.0632$) we mean, there is a very big difference. This explains that the Python language occupies the first position. This is due to the development of special compiler in its language and methods of writing the code.

In the table (7) we have explain the difference between programming languages in terms of degree of complexity, code size and code execution time

The comparison value for each part:

-Complexity degree (high, medium, normal)

-Code size (large, medium, small)

-The execution time is in milliseconds

Table(7) Comparative between programming languages

	Degree of complexity	Code size	Execution time
C language	high	Large	0.076 ms
C++ language	medium	Medium	0.135 ms
Java language	medium	Medium	2 ms
C# language	medium	Medium	7 ms
Python language	normal	Small	0.000997 ms

REFERENCES

- [1] Prajakta, G., Sumedh, P. "Smart Coding using New Code Optimization Techniques in Java to Reduce Runtime Overhead of Java Compiler" international Journal of Computer Applications, Volume 125 No.15,(2015), 11-16.
- [2] Kevin, W. Albert, N., Andreas, Gal., David, G. "Optimization Strategies for a Java Virtual Machine Interpreter on the Cell Broadband Engine" 1Trinity College Dublin, Dublin, Ireland, 2ETH Zurich, Zurich, Switzerland. 3University of California, Irvine, CA, USA, (2008).
- [3] Peter, S. "Numeric performance in C, C# and Java", IT University of Copenhagen Denmark, Version 0.9.1 of 2010-02- 19. (2010).
- [4] Guihot, H. "Optimizing Java Code. Pro Android Apps Performance Optimization", Springer, (2012), 1-31.
- [5] Matthieu A. "Teaching compilers with python", January 30, (2010), <http://www.hearc.ch/hearc/fr/isis/>
- [6] Halambi A., Grun P., Ganesh V., Khare A., Dutt N., Nicolau A. "EXPRESSION: A Language for Architecture Exploration through Compiler/Simulator Retargetability". In Proc. of the Conf, on Design Automation and Test Europe (DATE99), March 1999.
- [7] Mohammed, N.M., Lomte, S.S. "Secure and Efficient Outsourcing of Large Scale Linear Fractional Programming," Advances in Intelligent Systems and Computing, Springer, Vol. 1025, (2020).
- [8] AL-Bakhrani, A., Hagar, A., Hamoud, A., Kawathekar, S. "Comparative Analysis of Cpu Scheduling Algorithms: Simulation and its Applications". International Journal of Advanced Science and Technology, Vol. 29, No. 3, (2020), 483-494.