ANTHONY BABAJIDE BALOGUN
ID No: **UB73361SIN82521**

COURSE TOPIC:

PROGRAMMING LANGUAGES AND APPLICATIONS

ATLANTIC INTERNATIONAL UNIVERSITY
MAY 2022

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION

A programming language is defined as a set of symbols and rules for commanding or instructing a computer to carry out or to perform specific tasks. These types of symbols and rules are designed by some experienced and knowledgeable personnel known as programmers who have to follow all the specified rules before coding them using what is known as a programming language. As we may be aware, computer only understand what is known as machine language otherwise called binary numbers which is in form of O's and 1's.

An algorithm is defined as a step-by-step procedure or set of instructions for solving a problem or accomplishing a task. Every known computerized device makes use of algorithms to perform its functions. It makes life easier by reducing the time it takes to manually do things and also allow workers to be more proficient.

Moreover, a programmer must make use of the five basic parts of an algorithm to create a successful program. Firstly, he/she must describe or express the problem at hand in mathematical terms before creating the formulas and processes that creates results. Secondly, the programmer inputs the outcome parameters, and then he/she executes the program repeatedly to test its functionality and accuracy. The result is produced after the parameters has gone through the set of instructions in the program.

Furthermore, in this essay, I shall be writing on the Types of Programming Language; Categories of Programming Language; Steps to Designing Algorithms; Notations, Flowcharts and Pseudo Code including a program sample in one of the favourite programming languages called JAVA compared to a Pseudo Code. I will also write about programming languages and their applications.

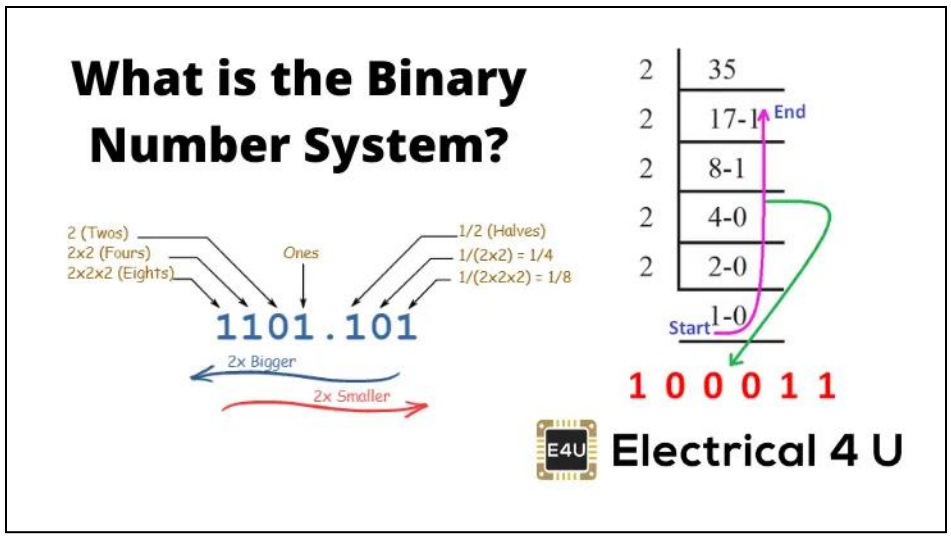
2.0 PROGRAMMIMG LANGUAGE

2.1 Definitions

A programming language is defined as a set of symbols and rules for commanding or instructing a computer to carry out or to perform specific tasks. These types of symbols and rules are designed by some experienced and knowledgeable personnel known as programmers who have to follow all the specified rules before coding them using what is known as a programming language. As we may be aware, computer only understand what is known as machine language otherwise called binary numbers which is in form of O's and 1's.

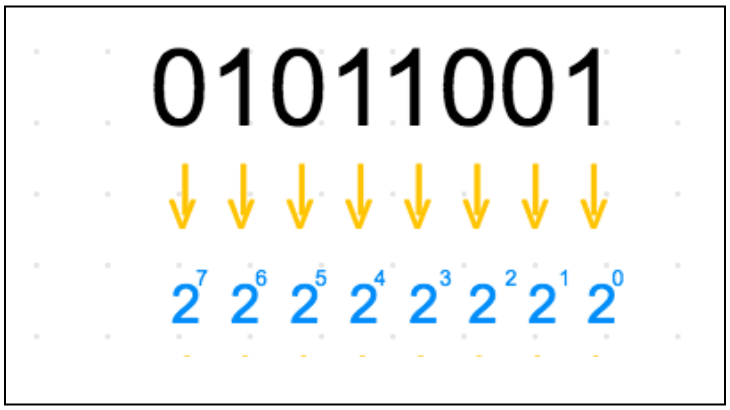
The English-like instructions or codes must first be converted into binary numbers before a computer can understand it, in other words, the user must communicate with the computer using the language (binary numbers) which it can understand.

As humans, we normally represent numbers in the decimal system. Therefore, counting the number from 1 to 10 is as simple as 1,2,3,4,5,6,7,8,9,10. On the other hand, Computers represent all information in bits. Computers use the binary number system in order to represent numbers with 0s and 1s. So, when a computer counts from 1 to 10, it looks like this: 0001, 0010, 0011, 0100, 0101, 0110, 1000, 1001, 1010, 0111.



(Electrical4U, 2020)

(Figure 1: What is the Binary Number System)



(Vivah, 2018)

(Figure 2: Binary Numbers)

3.0 TYPES OF PROGRAMMING LANGUAGE

Programming language is classified into the following types;

- Machine language
- Assembly language
- High level language

3.1 Machine Language

Computer only understand what is known as machine language otherwise called binary numbers which is in form of O's and 1's. In order to execute any program written in any programming language, the conversion to machine language is very essential. Conversion is not required for any program written in machine language as this can be executed directly on computer. The machine language program is absolutely translation free and it saves time because it's execution is pretty fast.

Its disadvantages include;

- Hard finding errors in the machine language
- Time consuming
- Machine dependent: program developed or written for a particular computer may not run on another type of computer

3.2 Assembly Language

Assembly language was developed in order to make the programming process easier. It is logically equivalent to machine language but the only added advantage is that, it is easier and more convenient for people to read, write and better to understand.

It uses symbolic notation to represent machine language instructions which are called low level language because they are relatively closer to the machines. An assembler translates assembly language instructions into a machine language which makes it easier to understand and use. Unlike machine language, errors are easily detected and located.

Assembly language comes with a few disadvantages though and they include;

- Machine dependent - the program which can be executed on a particular machine depends solely on the architecture of that particular computer.
- Assembly language is considered to be hard to learn
- Programmer needs to have the hardware knowledge to create applications
- It is less efficient
- Execution time is more than machine language program
- Assembler is needed for conversion to machine language

3.3 High level language

It consists of 'English-like words' and 'rules symbols' which are to be conformed with while coding a program. Furthermore, interpreter or compilers which translate high level language into machine language are required in order to convert these programs into machine readable form.

A compiler reads the whole program written in high level language at a go and translates it to machine language in lumpsum and if any error is encountered, such error will be displayed on the computer screen. Interpreter on the other hand reads and translates high level language program in line-by-line manner. It translates statement from a source code into a machine code at a time, and runs it before translating the next statement. The execution of the program is stopped when an error is encountered and such error message is displayed on the computer screen.

Its advantages include;

- Readability
- Easier to learn and understand
- Portability between machines.
- Easy debugging
- Easy to find and correct errors

Disadvantages

- Less efficient
- More execution times

4.0 CATEGORIES OF PROGRAMMING LANGUAGES

Programming language is classified into the following Categories:

- Interpreted programming languages
- Functional programming languages
- Compiled programming languages
- Procedural programming languages
- Scripting programming language
- Markup programming languages
- Concurrent programming languages
- Object oriented programming languages

4.1 Interpreted programming languages

Under this programming language, most of its instructions executes directly without previously compiling a program into machine language instructions. Examples include; Pascal and Python programming languages

4.2 Functional programming languages

This defines every computation bound to mathematical calculations as a mathematical evaluation. Examples include; Clean and Haskel programming languages

4.3 Compiled Programming languages

This is a programming language whose execution are typically run by compilers and not interpreters. It generates a machine code from the source code. Examples include; "C", "C++", "C#" and "JAVA"

4.4 Procedural programming languages

It specifies the steps that the programs should take in order to get to an intended state. Furthermore, it can be referred through a procedure call which makes the programs to be structured. Examples include; Hyper talk and MATLAB

4.5 Scripting languages

These are programming languages which controls an application and can execute independent of any other application. Mostly embedded in the application that they control and are used to automate the communication with external program. Examples include; Apple script and VB script

4.6 Markup languages

This is known as an artificial language that uses annotations to text that shows hoe the text is to be displayed. Examples include; HTML and XML

4.7 Concurrent programming language

It provides for the implementation of operation concurrently. This could be either within a single computer or across multiple systems. Examples include; Joule and Limbo

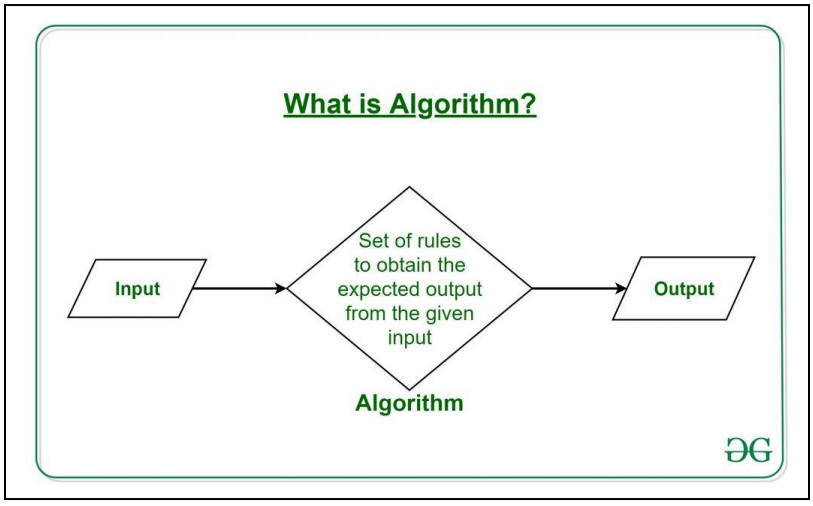
4.8 Object oriented programming language

This is paradigm based on the concept of objects which may contain data in the form of procedures commonly known as methods. Examples include; Lava and Moto

5.0 ALGORITHMS

An algorithm is defined as a step-by-step procedure or set of instructions for solving a problem or accomplishing a task. Every known computerized device makes use of algorithms to perform its functions. It makes life easier by reducing the time it takes to manually do things and also allow workers to be more proficient.

Furthermore, a programmer must make use of the five basic parts of an algorithm to create a successful program. Firstly, he/she must describe or express the problem at hand in mathematical terms before creating the formulas and processes that creates results. Secondly, the programmer inputs the outcome parameters, and then he/she executes the program repeatedly to test its functionality and accuracy. The result is produced after the parameters has gone through the set of instructions in the program.



(GeeksforGeeks, 2022)

(Figure 3: What is algorithm?)

5.1 Properties of Algorithms

- Should be written in simple English
- Should be precise and unambiguous
- Instructions should not be repeated infinitely
- Should conclude after a finite number of steps
- Should have an end point
- Derived results should be obtained only after the algorithm terminates

5.2 Qualities of Algorithm

- Time
- Memory
- Accuracy

5.3 Advantages of Algorithms

- It is easy to understand
- Step-wise representation of a solution to a given problem
- Problem is broken down into smaller pieces

5.4 Disadvantages of Algorithms

- Time-consuming
- Branching and Looping statements are difficult to show in Algorithms

6.0 STEPS TO DESIGNING AN ALGORITHM

The following pre-requisite are required in order to write an algorithm:

- Identify the problem this algorithm will solve
- Identify the constraints of the problem
- Determine the input to be accepted in order to solve the problem
- Consider the output to be generated after the problem has been solved
- Determine the solution to this problem

Let us consider the steps to be taken in a simple algorithm to add three numbers and then print the sum either on a paper as a hard copy or on the screen as a soft copy.

Step 1

- Identify the problem this algorithm will solve: Add 3 numbers and print their sum
- Identify the constraints of the problem: digits only
- Determine the input to be accepted in order to solve the problem: Add 3 digits
- Consider the output to be generated after the problem has been solved: addition of 3 digits accepted as the input
- Determine the solution to this problem: adding 3 digits by making use of the plus (+) operator
- Consider the algorithm to be written in order to solves the problem: algorithm to be written in 3 different programming languages

Step 2

Designing the algorithm with the help of above pre-requisites

START

- Declare 3 integer variables **n1, n2 and n3**
- Accept the three numbers to be added as inputs in variables **n1, n2 and n3**
- Declare integer variable **res** to store the summation of the 3 numbers
- Add the 3 numbers and then store the result in the variable **res**
- Then print the value of variable **res**

END

Step 3

In order to test the algorithm, I will write codes to implement it in three different programming language namely; ***C, C++ and Python3***

6.1 C Programming Language

// C program to add three numbers with the help of the above designed algorithm in step 2

```
#include <stdio.h>
int main()
{
    // Variables to take the input of the 3 numbers
    int n1, n2, n3;
    // Variable to store the resultant sum
    int res;
     // Accept the 3 numbers as input
     printf("Enter the 1st number: ");
    scanf("%d", &n1);
    printf("%d\n", n1);
    printf("Enter the 2nd number: ");
    scanf("%d", &n2);
    printf("%d\n", n2);
    printf("Enter the 3rd number: ");
    scanf("%d", &n3);
    printf("%d\n", n3);
```

```
    // Calculate the sum using the plus (+) operator
   // and store it in variable res
     res = n1 + n2 + n3;
   // Print the sum
     printf("\nSum of the 3 numbers is: %d", res);
       return 0;
}
```

Output

```
Enter the 1st number: 10
Enter the 2nd number: 30
Enter the 3rd number: 20

Sum of the 3 numbers is: 60
```

6.2 C++ Programming Language

// C++ program to add three numbers with the help of above designed algorithm in step 2

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
 // Variables to take the input of the 3 numbers
   int n1, n2, n3;
 // Variable to store the resultant sum
   int res;
 // Accept the 3 numbers as input
   cout << "Enter the 1st number: ";
   cin >> n1;
   cout << " " << n1 << endl;
   cout << "Enter the 2nd number: ";
   cin >> n2;
   cout << " " << n2 << endl;
   cout << "Enter the 3rd number: ";
   cin >> n3;
   cout << " " << n3;
```

```
        // Calculate the sum using the plus (+) operator and then store it in variable
           res
           res = n1 + n2 + n3;
        // Print the sum
           cout << "\nSum of the 3 numbers is: "
           << res;
          return 0;
}
```

Output

```
Enter the 1st number: 10
Enter the 2nd number: 30
Enter the 3rd number: 20

Sum of the 3 numbers is: 60
```

6.3 Python3 Programming Language

```
# Python3 program to add three numbers with the help of above designed
  Algorithm in step 2

    if __name__ == "__main__":
    # Variables to take the input of the 3 numbers
    n1 = n2 = n3 = 0

    # Variable to store the resultant sum
    res = 0

    # Accept the 3 numbers as input
     n1 = int(input("Enter the 1st number: "))
     n2 = int(input("Enter the 2nd number: "))
     n3 = int(input("Enter the 3rd number: "))
     # Calculate the sum using the plus (+) operator and store it in variable sum
     res = n1 + n2 + n3

     # Print the sum
     print("\nSum of the 3 numbers is:", res)
```

Output

Enter the 1st number: 10

Enter the 2nd number: 30

Enter the 3rd number: 20

Sum of the 3 numbers is: 60

7.0 PROGRAMMING LANGUAGES AND APPLICATIONS

7.1 Python

Python is a high level and general-purpose programming language which was released in the year 1991 with a version known as Python 0.9.0. In year 2000, 'Python 2.0' was released with new features until ‘Python 3.0' was released in 2008 after Python version 2.0 was discontinued.

All versions of Python which include; 2.7, 3.8.8 and 3.9.2 were discovered to have security issues which could lead to remote code execution and web cache poisoning, so they were expedited. The latest version of Python in the market as at the time of writing this essay is, Python 3.11.0a6 which was released in February 2022.

7.1.1 Applications for Python

(i) Web

Many application domains make use of Python and it also offers various options for web development framework such as; Django and Pyramid. Moreover, Flask and Bottle which are known as Micro-frameworks also make use of Python.

The applications for Python are also used in Advanced content management systems which include; Plone and Django CMS. Its library supports many Internet protocols such as;

- HTML and XML
- JSON
- E-mail processing
- FTP
- MAP

(ii) Scientific and Numeric Computing

Python is used both in the scientific and numeric computing. Its library contains some features which makes this possible and they include;

- **SciPy** - collection of packages for engineering, sciences and mathematics
- **Pandas** - used for data analysis and data modeling
- **IPython** - used for editing and recording of a work session, visualizations and parallel computing

(iii) Software Development

- Used for building controls and testing within applications
- Kivy - for coding and programming multitouch applications
- Used for building ERP and e-commerce systems and applications
- Used for bug tracking
- Used in project management

7.1.2 Sample Program

```
# Python3 program to add three numbers

if __name__ == "__main__":
# Variables to take the input of the 3 numbers
n1 = n2 = n3 = 0

# Variable to store the resultant sum
res = 0

# Accept the 3 numbers as input
 n1 = int(input("Enter the 1st number: "))
 n2 = int(input("Enter the 2nd number: "))
 n3 = int(input("Enter the 3rd number: "))
```

```
# Calculate the sum using the plus (+) operator and store it in variable sum
res = n1 + n2 + n3

# Print the sum
print("\nSum of the 3 numbers is:", res)
```

Output

Enter the 1st number: 10
Enter the 2nd number: 30
Enter the 3rd number: 20

Sum of the 3 numbers is: 60

7.2 JavaScript

JavaScript which is often abbreviated JS alongside HTML and CSS are all programming languages that forms the core technologies behind the World Wide Web (WWW). Many web browsers have improvised a JavaScript engine in order to execute the code on users' devices.

More so, over 97%of websites uses JavaScript on the client side for web page execution by making use of third-party libraries. JavaScript is a high-level that conforms to the ECMAScript standard. It is also multi-paradigm which supports event-driven and imperative programming styles.

Furthermore, it has APIs (Application Programming Interfaces) that enables working dates, text, expressions, data structure and Document Object Model (DOM). Initially, its engines were used only in web browsers, but they have now formed a core component of various servers and applications. The most common and well-known runtime system for this type of usage is Node.js.

However, Javascript should not be confused with Java programming language because, although they have similar name, standard libraries and syntax but they are distinctively different in design

7.2.1 Applications for Javascripts

(i) Web Applications

Javascripts is used to code a well robust web applications such a Google Maps. We can notice that, in Google Maps, user needs to just click and drag the mouse and immediately, the details are displayed. The use of JavaScript forms the concepts behind this type of operations.

(ii) Web Development

JavaScript is a tool used to create web pages because it enables the addition of special effects and dynamic behavior to the web pages.

Javascripts is also used for validation purposes on websites, helps to execute complex actions, allows visitors to interact with the websites and it allows the content of a document to be loaded without necessarily reloading the webpage itself.

(iii) Mobile Applications

Mobile devices are broadly used to access the internet nowadays. This is made possible by using JavaScript to build an application for non-web contexts. The features in JavaScript have made it a powerful tool for creating mobile applications which can operate on different operating systems. Therefore, there is no need to write different codes for iOS and Android operating systems because, codes are needed to be written once and it then runs on different platforms.

(iv) Gaming

JavaScript has different types of libraries and frameworks for creating different types of game which could either be in 2D or 3D. Such engines in Javascript that allows the programming of web games include; PhysicsJS and Pixi.js. Web Graphics Library (WebGL), which is the JavaScript API is also needed to render 2D and 3D images on most web browsers.

(v) Presentations

"Libraries such as RevealJs and BespokeJs" are used to create a web-based slide deck for presentations which works well with mobile devices and tablets.

(vi) Server Applications

Web applications usually have a server-end connection. So, when request is made, JavaScript generates the content and handles HTTP requests which runs on the server through Node.js which serves as an environment that contains all the necessary tools needed for JavaScript to execute on servers.

(vii) Web Servers

As described above, web server can be created by using Node.js which are fast and doesn't use buffering and transfer chunks of data. So, when a user makes a request, the HTTP server will show HTML which will be included in the HTTP header.

7.2.2 Sample Program

Sample code to find the lowest common multiple (LCM) of two numbers

```
/* Finds the lowest common multiple (LCM) of two numbers */

function LCMCalculator(x, y) {
  const checkInt = function(x) {
   if (x % 1 !== 0)
  throw new TypeError(x + "is not an integer");
 return x;
 };

 this.a = checkInt(x)
 this.b = checkInt(y);
}

LCMCalculator.prototype = {
 constructor: LCMCalculator,
 gcd: function() {
 let a = Math.abs(this.a), b = Math.abs(this.b), t;



 if (a < b) {

 }

 while (b !== 0) {
  t = b;
  b = a % b;
  a = t;
 }

this['gcd'] = function() {
return a;
 };

 return a;
},

"lcm": function() {

let lcm = this.a / this.gcd() * this.b;
```

```
this.lcm = function() {
return lcm;
};

return lcm;
},

toString() {

return `LCMCalculator: a = ${this.a}, b = ` + this.b;
    }
};

function output(x) {
document.body.appendChild(document.createTextNode(x));
document.body.appendChild(document.createElement('br'));
}

[
 [25, 55],
 [21, 56],
 [22, 58],
 [28, 56]


].map(function(pair) {
 return new LCMCalculator(pair[0], pair[1]);
}).sort((a, b) => a.lcm() - b.lcm())
forEach(printResult);

function printResult(obj) {
output(obj + '', gcd = '' + obj.gcd() + '', lcm = '' + obj.lcm());
}
```

The output result displayed in the browser window should look like this:

LCMCalculator: a = 28, b = 56, gcd = 28, lcm = 56
LCMCalculator: a = 21, b = 56, gcd = 7, lcm = 168
LCMCalculator: a = 25, b = 55, gcd = 5, lcm = 275
LCMCalculator: a = 22, b = 58, gcd = 2, lcm = 638

7.3 JAVA

The Java programming language is a general-purpose programming language that is class-based, object-oriented and developed similar to C++ programming language sharing most of the syntax with it. It was developed inside Sun Microsystems by James Gosling and was released in May 1995.

It is structured in a way that; software developers can write code anywhere and run it anywhere without getting worried about the computer architecture. Java is also referred to as **WORA** which means; Write Once, Run Anywhere.

This further means that, Java code when compiled can run once and can then be run on all platforms that executes Java Runtime Environment without making use of any compiler to recompile.

7.3.1 Applications for Java

Java can be used by developers to build applications that run JRE and supports applications that run on a single device like a desktop or mobile phone for various platform where developers are able to write Java-based applications for any of these platforms. Likewise, computer servers, desktops, mobile phones, tablets and web browsers all make use of Java. It can also be used to write application applets for web pages.

Java is used for the following:

- GUI applications
- Web servers and applications servers
- Middleware applications
- Web applications
- Mobile applications
- Embedded systems
- Enterprise applications

7.3.2 Benefits of Java

- Simple and easy to learn
- Java shares syntax with C and C++
- Object-oriented programming language
- Multithreading is supported by Java
- Platform agnostic language

7.3.3 Disadvantages of Java

- Applications must be run on JRE.
- UIs built using Java are less attractive
- No backup facility
- Garbage collectors provided with Java are automatic

7.3.4 Best Practices for Using Java

- Code in a safe environment
- Plan object requirements before writing code
- Follow naming conventions
- Avoid the potential for memory leaks
- Avoid empty catch blocks
- Avoid using loops with indexes
- Reserve sufficient memory
- Re-scale images to use less resources

7.3.5 Sample Program

This is a simple program to multiply two numbers in Java. This program accepts two integers, multiply them and then print their product

```
import java.util.Scanner;

public class Demo {

   public static void main(String[] args) {

      /* This reads the input provided by user
       * using keyboard
       */
      Scanner scan = new Scanner(System.in);
      System.out.print("Enter first number: ");

      // This method reads the number provided using keyboard
      int num1 = scan.nextInt();

      System.out.print("Enter second number: ");
```

```
        int num2 = scan.nextInt();

        // Closing Scanner after the use
        scan.close();

        // Calculating product of two numbers
        int product = num1*num2;

        // Displaying the multiplication result
        System.out.println("Output: "+product);
    }
}
```

Output:

```
Enter first number: 15
Enter second number: 6
Output: 90
```

7.4 C Programming Language

C is a procedural programming language with a static system created at Bell Laboratories in the early 1970s by Ken Thompson and Dennis Ritchie. C programming language is better known as a machine-independent programming language used for creating many types of applications and operating systems such as Windows, databases such as; Oracle database, Git, Python interpreter, and games.

Operating systems and other application software designed for computer architectures all makes use of C programming language.

C is known to be very fast with a large library and it is considered to be one of the oldest and most efficient programming languages which is extensively make use of all over the world.

7.4.1 Applications for C

- Operating System
- 3D Movies
- Intermediate Language
- Aid the Development of New Programming Language
- Embedded Systems

7.4.2 Benefits of C

- Portability and Efficiency
- Memory Manipulation
- Resources Are Used in a Definite Way

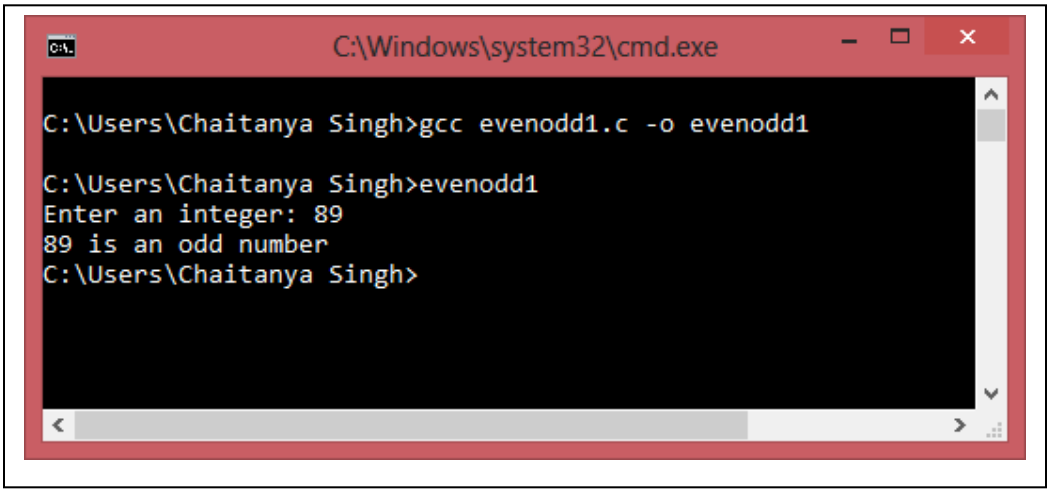
7.4.3 Sample Program

This is a sample program to check whether the input integer number is even or odd using the modulus operator (%)

```
#include<stdio.h>
int main()
{
```

```
// This variable is to store the input number
   int num;

   printf("Enter an integer: ");
   scanf("%d",&num);

   // Modulus (%) returns remainder
   if ( num%2 == 0 )
      printf("%d is an even number", num);
   else
      printf("%d is an odd number", num);

   return 0;
}
```

Output result



7.5 C# Programming Language

C# is a general-purpose programming language developed in the year 2000 by Microsoft's Anders Hejlsberg, a Danish software engineer. It is an object-oriented programming language (OOP) primarily used on the Windows .NET framework and it can also be applied to an open-source platform.

It was initially designed to compete with Java and judging by the rise to popularity amongst the seasoned developers, it is safe to admit that the goal has been achieved.

7.5.1 Applications for C#

C# is relatively considered to be a great choice for developers with moderate to advanced experience with coding because it is fairly simple to grasp and excel in. Once you are introduced to C# and you dedicate your time on it, you are expected to advance very quickly from a beginner to an expert within a short space of time. This is made possible because, C# is a high-level language and it is easier to read and write.

C# programming language can be used to develop the following applications:

- Website development
- Windows applications
- Games

Programs and applications written in C#:

- Windows Installer XML
- Microsoft Visual Studio
- Paint.NET
- Open Dental
- KeePass
- FlashDevelop
- Banshee
- NMath
- Pinta
- OpenRA

7.5.2 Advantages of C#

- C# can save you time
- There’s a low learning curve
- C# is easy to maintain
- There’s a large community—very large
- C# is object-oriented
- C# comes with plenty of career opportunities

7.5.3 Sample Program

This is a simple C# Program to check if a given number is either odd or even

number

```
using System;

public class EvenOddProgram
{
        public static void Main()
        {
                int num;
         Console.Write("Enter a Number : ");

        num = int.Parse(Console.ReadLine());

                //divide it by 2, if remainder is 0 then it is even number
         if (num % 2 == 0)
         {
            Console.WriteLine(num+" is an Even Number");
         }
         else
         {
            Console.WriteLine(num+ " is an Odd Number");
         }
        }
}
```

Output:

```
Enter a Number : 5
5 is an Odd Number
```

7.6 C++ Programming Language

C++ is a general-purpose programming language created by Bjarne Stroustrup in 1979 at Bell Labs in Murray Hill, New Jersey. It is regarded as an improvement over C language designed with an orientation towards large systems and resource-constrained software.

C++ has grown significantly over time with latest version C++20 which comprises of object-oriented and generic features in addition to its low-level memory manipulation. Furthermore, C++ is super scalable and this enables developers to have control over how their applications make use of resources.

7.6.1 Application for C++

Almost all the programs, be it Windows, a web browser, games or a video editing software codebase is written either in C or C++. Therefore, the uses of C++ are evident in almost all applications in use today.

The following applications or software are written in C++ language:

- Operating Systems
- Games
- GUI Based Applications

- Web Browsers
- Embedded Systems
- Banking Applications
- Compilers
- Database Management Software
- Cloud/Distributed Systems
- Libraries
- Switches
- Advanced Computation and Graphics

7.6.2 Advantages of C++

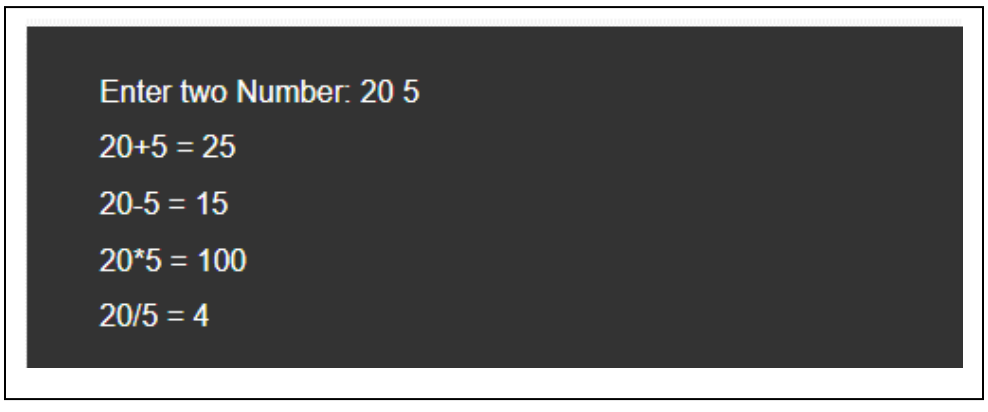
- Object-Oriented
- Speed
- Compiled
- Rich Library Support
- Pointer Support
- Closer to Hardware

7.6.3 Sample Program

This is a simple C++ program to add, subtract, multiply and divide two numbers:

```
#include<iostream>
using namespace std;
int main()
{
  // declare variables
  double num1, num2;

  // take input from end-user
  cout << "Enter two Numbers :: ";
  cin >> num1 >> num2;

  // addition of two number
  cout << num1 << "+" << num2 << " = "<< num1+num2 << endl;

  // subtraction of two number
  cout << num1 << "-" << num2 << " = "<< num1-num2 << endl;

  // multiplication of two number
  cout << num1 << "*" << num2 << " = "<< num1*num2 << endl;

  // division of two number
  cout << num1 << "/" << num2 << " = "<< num1/num2 << endl;

  return 0;
}
```

Output:-



8.0 CONCLUSION

To be a successful software developer or programmer, an individual must possess the skills of logical thoughts. This however takes considerable amount of practices with a self-motivation much like a hobby. In my own opinion, today's software engineering is much greater compared to that of about twenty years ago. Presently, not only must you program an algorithm successfully, a Windows interface must also be taken into consideration and well designed.

Moreover, teaching and learning programming on a personal computer requires a Windows development environment which include; Visual Basic tools, C, C++, or even FORTRAN. Visual Basic, C#, Python presents the best and the most convenient environment to design Windows application interfaces in today's programming world.

Consequently, many students are now being taught using various commercial programs such as Microsoft’s Excel spreadsheet program. Matlab is another popular commercial program which requires students to execute and implement pseudo programming steps in order to obtain an answer to a problem.

Furthermore, how can an individual be motivated to become a software developer in this highly modern computer environment?

I am particularly concerned about the fact that; the number of talented software developers is diminishing whilst the reliance on commercial and Windows-based software grows.

Our dependency on commercial software today is too heavy and I am beginning to wonder, is there is a pool of talented programmers who have developed their skills through personal learning somewhere?, but I am nowhere closer to any answer to this important question.

9.0 Bibliography

Electrical4U. (2020, October 22). *Binary Number System: What is it? (Definition & Examples)*. Retrieved from What is the Binary Number System?: https://www.electrical4u.com/binary-number-system-binary-to-decimal-and-decimal-to-binary-conversion/

GeeksforGeeks. (2022, January 13). *Introduction to Algorithms*. Retrieved from What is Algorithm? Algorithm Basics: https://www.geeksforgeeks.org/introduction-to-algorithms/

Geeksforgeeks.org. (2021, Sept 30). *How to write a Pseudo Code?* Retrieved from How to write a Pseudo Code?: https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/

Smartdraw. (n.d.). *Common Flowchart Symbols*. Retrieved from Flowchart Symbols: https://www.smartdraw.com/flowchart/flowchart-symbols.htm#:~:text=Flowcharts%20use%20special%20shapes%20to,are%20known%20as%20flowchart%20symbols.

Smartdraw. (n.d.). *Flowchart Symbols*. Retrieved from Flowchart Symbols: https://www.smartdraw.com/flowchart/flowchart-symbols.htm#:~:text=Flowcharts%20use%20special%20shapes%20to,are%20known%20as%20flowchart%20symbols.

Vivah, L. (2018, June 22). *Learn How to Read Binary in 5 minutes*. Retrieved from Learn How to Read Binary in 5 minutes: https://medium.com/@LindaVivah/learn-how-to-read-binary-in-5-minutes-dac1feb991e