FROM C TO C++, C#, AND JAVA

Hieu Vu
American University of Nigeria
98 Lamido Zubairu Way
Yola by Pass, P.M.B. 2250,
Yola, Adamawa State, Nigeria

Abstract
Computer technology changed rapidly, programming languages also evolved with the time. The C language became a popular language and established itself as the foundation for the newer, more powerful programming languages that were developed later such as C++ (pronounce C plus plus), C# (pronounce C-sharp) and Java. The later three languages also accommodate new programming paradigm “Object-Oriented Programming (OOP), that gain widespread quickly. Now the programmers can write larger applications efficiently satisfied the demands.

Keywords:
Programming, C language, C++, Java, C#, Object-oriented

I. INTRODUCTION
Today, technology improves rapidly in both hardware and software development. Programming languages also changed over the time, allowing computer scientists, programmers to write programs, develop software applications more efficiently. C language was developed at Bell labs by Ken Thompson and Dennis Ritchie in the early 1970s. First, it was used to rewrite UNIX that resulted in a more popular, portable operation system. Gradually, C language became the most common, language of choice for writing operation systems, UNIX, Microsoft Windows, Mac OS-X, and GNU Linux were all written in C. Not only the operating systems, C language was also used to write other high level language compilers: Perl, Python, PHP, ..[1]. This paper presents special features of the C language then the new enhanced features of newer languages: C++, C#, and Java. We might think, C started a new era for programming in both programming language and programming paradigm.

II. WHY C LANGUAGE?
C is flexible, offers a convenience way of writing program such as (++, --) auto-increment / decrement operators. A block of statements in C language are enclosed inside a pair of braces ({ }) instead of between two keywords “BEGIN and END”.
Example:

```
i = i + 1;          //C language (Shorthand style)
i += 1;             //C language (auto-increment)
```

or
```
i++;                 //C language (auto-increment)
```
Example:

Look at the following program segments of “Binary Search” an array

(C language)                                                  (Pascal)
int n, key;     var low, mid, high: index;
{
    result: integer;
    int low = 0, high = n-1, mid;    begin
    while(low <= high)                   low := 1;
    {                                    high := n;
        mid = (low + high) / 2;           while(low <= high) do
        if(array[mid] > key)              begin
            high = mid - 1;                   mid := (low + high) / 2;
        else if(array[mid] < key)            if array[mid] < key then
            low = mid + 1;                       high := mid – 1
        else                                 else if array[mid] > key then
            return mid;                          low := mid + 1;
        }                                      else
    }                                          else
    return -1;                                  result = mid;
}                                              end;
}                                                   return -1;
    result = mid;
}                                                      } [2]

Another special feature of C, it is classified as a high level language but it also has a “closed look” on the hardware, manipulation at the bit level such as the Assembly (the one and only one low level language). The operators that perform operation on bit-by-bit called “Bitwise operators”, these included: & (and), | (or), ^ (xor – exclusive or) and ~ (not).

Example: In ASCII characters:

var_1 = 00100110
var_2 = 00110011
then:

var_1 & var_2 = 00100010      (both must have value of 1)
var_1 | var_2 = 00110111      (one value of 1 is enough)
var_1 ^ var_2 = 00010101      (cannot have the same value)
~var_1         = 11011001      (invert all bits of var_1)

Another interesting operation on bit is the (>>) shift-right, (<<) shift-left. We know, in binary, everything is a multiple of \(2^n\), hence:

if \(x = 00000100\) (binary) or 4(decimal)

then:

\(x >>= 2 \Rightarrow x = 00000001\) or just 1 (decimal – divided by 4)
\(x <<= 2 \Rightarrow x = 00010000\) or 16(decimal – multiply by 4) [3]
III. FROM C TO C++

C++ is a general-purpose programming language developed by Bjarne Stroustrup, a computer scientist from Bell Labs in the 1980s. C++ is regarded as an extension of the C language and became one of the most popular programming languages in the 1990s. Most of C language included in C++ and C++ compiler can compile source program written in C. The first and most important feature that added to C language is the concept of Object-Oriented, this concept opened a new door for new programming paradigm “Object-Oriented Programming” that has been using up to today. Later, C++ added more features such as: Virtual Function and Polymorphism, Operator Overloading, Templates, and Exception Handling...[4].

III.1. VIRTUAL FUNCTIONS and POLYMORPHISM

In C++, when both super class and sub class have functions with the same names such as “print()”, to display the data of the objects. These functions only display data that common by both objects of super and sub classes. This problem called “Slicing problem”, the one that belongs to sub class is simply sliced off. To correct this problem, the function in super class should be declared as “virtual”.

Example: Slicing problem

class A{       //super class
    public:
        void display(){
            cout << “From base class A.\n”;
        }
};
class B: public A{   //sub class derive from A
    public:
        void display(){
            cout << “From derived class B. \n”;
        }
};

int main(){       //main function that uses the two classes
    A *a;       //pointer to object A
    B b;        //object B
    a -> display(); //point to display function
    a = &b;    //assign address of b to pointer
    a -> display(); //intended to display object b
    return 0;
}

Output:
From base class A.
From base class A. (Only display() function in super class is invoked)

To correct this problem, we need to declare function display() in super class “virtual”. Then re-run the program.

class A{       //super class
III.2. OPERATORS OVERLOADING

Operators overloading means redefining the operation of the operators. The C++ programmers can overload most of the built-in operators in C++. Overloaded operators are functions with the keyword “operator” and followed by the symbol of the operator being redefined. After overloaded, the operators will perform the tasks according to instructions in the function operator overloading.

Example:

```cpp
class Box{
    private:
        double width; //width of the box
        double length; //length
        double height;
    public:
        Box(double w, double l, double h){ //value constructor
            width = w;
            length = l;
            height = h;
        }
        void setWidth(double w){ //set new width
            width = w;
        }
        void setLength(double l){ //set new length
            length = l;
        }
        void setHeight(double h){ //set new height
            height = h;
        }
        double calVolume(){ //Calculate volume
            return width * length * height;
        }
    //Overload “+” operator to add two Box objects
    Box operator+(const Box& b){
        Box box; //instance Box object b
        box.width = this->width + b.width;
        box.length = this->length + b.length;
        return box;
    }
};
```
box.height = this->height + b.height;
    return box;
};

int main(){       //main function to use class Box
    Box b1 = new Box(6.0, 7.0, 5.0);       //Box b1
    Box b2 = new Box(12.0, 13.0, 10.0);    //Box b2
    Box b3 = new Box(0, 0, 0);
    cout << “Volume Box b1: “ << b1.calVolume() << endl;
    cout << “Volume Box b2: “ << b2.calVolume() << endl;
    //Add two Box objects
    b3 = b1 + b2;
    cout << “Volume Box b3: “ << b3.calVolume() << endl;
    return 0;
}

Output:
Volume Box b1: 210
Volume Box b2: 1560
Volume Box b3: 5400    (not equal 210 + 1560) [6]

 IV. FROM C, C++ TO C#

In year 2000, .NET (dot NET) revolutionized the way for programmers to build Web and Windows applications. C# (pronounce C-Sharp) was developed by Microsoft’s R&D team, led by Anders Hejlsberg and Scott Wiltamuth, it was the language of choice for .NET platform, built from lessons learned from other older languages: C, C++, and Java. C# offers many new features:
- C# brings back support for “struct” (structure) concept that was missing in Java. A struct is a restricted, lightweight type that can’t inherit or be inherited from a class but a struct can implement an interface.

Example:

```csharp
    public struct location{       //define struct location
        public int x{get; set;}
        public int y{get; set;}
        public override string ToString(){
            return (String.Format(“{0}, {1}, x, y));
        }
    }[7]
```

- C# provides support for “delegate”, an indirect technique for methods invocations. A delegate can be explained as a pointer to a method in C++

Example:

1. Without using delegate

```csharp
    class Controller{                //Without using delegate
        private FoldingMachine folder;    //Different machines
```
private WeldingMachine welder;
private PaintingMachine painter;
...
public void ShutDown(){           //Shutdown machines
    folder.StopFolding();
    welder.FinishWelding();
    painter.PaintOff();
}
...
}

2. Using delegate
class Controller{                   //Using delegate
delegate void stopMachineryDelegate();
private stopMachineryDelegate stopMachine;
...
public Controller(){
    this.stopMachine += folder.StopFolding;
    this.stopMachine += welder.FinishWelding;
    this.stopMachine += painter.PaintOff;
}
...
}

- C# also provides support for “Lambda Expression”, a new feature which is an expression that returns a function (method). Lambda Expression using syntax and semantics of the Lambda Calculus such as in the following examples.

Examples:
x => x * x                 //Symbol “=>” is Lambda Expression in C#
x => {return x * x;}       //Same as above
(int x) => x / 2
() => folder.StopFolding() //Calling a method
(x, y) => {x++; return x / y}

- Other new features of C# included:
1. LINQ (Language Integrated Query) that similar to SQL (Structured Query Language) used to access database.

    IEnumberable<string> customerFirstNames =
    customer.Select(cust => cust.FirstName);
    foreach (string name in customerFirstNames)
    {
        Console.WriteLine(name;
    }
From the example above, we can see the “Select” verb of SQL and also a new version of “for loop” in C#, foreach (    …    )

2. ADO.NET that can be used to fetch and display data from a Microsoft SQL server database
   try{
            //Fetch data and display order
            ...
            while(dataReader.Read()){
                //Code to display the current row
            }
   }

3. ASP.NET (Active Server Page) for development of Web applications. ASP.NET supports HTML documents.
   <asp:TextBox BackColor=”Blue” ForeColor=”White” Runat=”Server” />
   <asp:Label BackColor=”White” ForeColor=”Blue” Runat=”Server” Font-Bold=”True” />[8]

V. JAVA
Two important factors that influenced computer language are: improvements in the art of programming and the computing environment. Java is no exception, must satisfied the two factors before becoming the most popular, widespread programming languages. It was developed at Sun Microsystems in 1991, by a team led by James Gosling. The new language was initially named “Oak” then renamed “Java” in 1995.

Unlike the other older languages, that was designed to run on a particular machine (CPU), Java is portable language that can run on multi-platforms (cross-platforms), the Internet and World Wide Web (WWW).

V.1. JAVA and C, C++
Java statements are very much the same as C, C++ statements, we can say Java inherits its syntax from C and concept of object-oriented from C++. An example of adopting C’s syntax can be found in input/output statements: “scanf” and “printf”. In the early years of Java, to input data from the keyboard, we need to download class keyboard from the Internet, later we have class “Scanner” for reading an input stream. The same for output, we used class “DecimalFormat” to format numeric values, now it is convinience and efficiently to use the printf statement.

Example:
   Scanner input = new Scanner(system.in);//instance of class Scanner
   int intNumber = input.nextInt();       //Read integer value
   System.out.printf(“%4d", intNumber);   //use 4 spaces to print

An important note that even influenced by C, C++, Java is not an extension or enhanced version of C++. It was designed for some special purposes not for replacing C++.
V.2. JAVA and C#

Microsoft developed C# as the leading language on .NET platform a few years after the release of Java. Both Java and C# inherit syntax and features from C, C++, but C# is closer to the “C family” and could be regarded as an extension of C and C++. In some aspect, C# refines Java statements make them more convenience to code. One of the requirement to create a Javabeans is each data member of the class must has a pair of “get and set” methods, C# refines by combining them together into one.

Example:

Java:
```java
public class Number{      //Java class Number
    private double number; //Data member
    ...
    public double getNumber{ //Get the number
        return number;
    }
    public void setNumber(double n){ //Set new value
        number = n;
    }
    ...
}  //End Java class Number
```

C#:
```csharp
public class Number{       //C# class Number
    private double number;  //Data member
    ...
    public double Num{      //Get and Set method for number
        get{                 //Get number
            return number;
        }
        set{                 //Set new value
            number = value;
        }
    }
    ...
}[[10]
```

VI. CONCLUSION

We are living in a changing world and it changes rapidly in science and technology. Computer programming is a very important skill for computer scientists and programmers as the demand for more new applications increase. We might think, programming is an art of problem solving using computer, however, without a suitable programming language, the programmer’s jobs would be more difficult, just like an artist without a right tool.

From this paper, we can see the evolution of programming languages over the time, the new language would inherit characteristics of the ancestors then adding new or enhanced features.
REFERENCES: