**Assignment 2**

1. (20 marks) Consider the following list of languages:

\* FORTRAN (FORmula TRANslation)

\* BASIC (Beginner’s All-purpose Symbolic Instruction Code)

\* COBOL (COmmon Business-Oriented Language)

\* Algol (Algorithmic Language)

\* C

\* C++

For each language, briefly discuss two of its features or functions that influence the development of programming languages

Ans: Some of the points described are the features or functions of the programming language which influenced the development of programming languages.

FORTRAN:

* It had included Dynamic allocation of values and pointers which is used to manipulate memory addresses.
* It introduced user defined data types which is very common these days.

BASIC:

* First ever introduction to loops. The most important feature that is used in day to day life of a program.
* User input from the keyboard was also added in the latter version of BASIC.
* User defined function was also introduced.

COBOL:

* Robustness of this language was its main attraction. It also had the most structured language which helped keeping the code organized.

Algol:

* It was a high level language in current time we can compare it to Java or C#.
* Code block was a thing which uplifted Algol. Which also helped introduce nested functions.

C :

* Combination of assembly language with high level is what C is. The code of C is fast, easy to understand and changes comparatively.
* It is also more portable than its ancestors. Once a code is written in a machine it can be run easily in another machine with very less prerequisites.

C++ :

* Encapsulation and Privacy of data is what makes C++ different from C. Although OOP was introduced before the DATA Privacy was a drastic change.
* Inheritance was also introduced which enhanced OOP.

2. (20 marks) Search the Web to find information on programming language PASCAL. Based on your findings, provide a brief analysis of PASCAL by considering the following points:

a. History of PASCAL

b. Main features of PASCAL that distinguish it from other languages.

c. The roles played by PASCAL.

d. The reasons for the rise and fall of PASCAL.

Note: the expected length of this short essay is about two pages (double-spaced).

Ans: The main Source of this information is [Wikipedia](https://en.wikipedia.org/wiki/Pascal_(programming_language)).

It was designed by Niklaus Wirth and first appeared in 1970. The programming language was named after a French mathematician, philosopher and physicist Blaise Pascal. It is an imperative which means a statement based manipulation of program state and procedural language. The language was developed to develop reliable programs which are efficient and also for the teaching of programming as systematic discipline. It was influenced by ALGOL and Simula and the main focus was to encourage good programming practice. It includes many constructs of Algol. The ISO 7185 was the PASCAL Standards published in 1983. Dr Wirth was one of the members in developing ALGOL and had intention to improve it but going forward his request was never considered. He decided to create a new programming language PASCAL with those features. Latter Apple used PASCAL to release

PASCAL has a lot of popularity in the teaching and academics arena because of its Easy to learn syntax and more closer to human language statements. According to Dr Wirth, programs are nothing but a combination of Algorithms and DATA Structures. It is structured language so the code is very organized and was easy to be an industry standard. It is famous for its days creating a transparent and efficient program. It was supported by most of the popular machines. So the code had very less prerequisites. PASCAL is a very strongly typed language it means that the syntax has fewer varieties to be written in. Makes it easier and has strong rules to understand and interpret. It also has better error handling functionality. The extensive error checking makes the program less buggi. It supports arrays, lists, external files and sets as Data types. Using functions and procedures it offers Structured programming. And the most overpowered feature, it supports OOP (Object Oriented Programming).

The first PASCAL compiler was designed in Zurich for the CDC 6000 series mainframe computer family. The compiler was written in C-Like language Scallop and then translated to PASCAl. So whenever there is a newer version the compiler compiles itself to upgrade. Only GNU Pascal compilers are written in C. For the object Pascal and Turbo Pascal, Apple Computer created its own Lisa Pasca for the Lisa workshop in 1982. This was a greatest influence in object Pascal creation.

Although the language was very popular in the 1980s to early 1990s the initial Dr Wirth’s definition of the language was questioned outside the scope of teaching as it had some major issues like array size and string lengths were a mandatory to define a variable and they cannot be passed as parameters. PASCAL was criticised for its poor library support, unpredictable boolean expression, lack of static variables and many other small issues. One of these issues also include that PASCAL was not supporting the functions to be passed as parameters to predefine the expected types of their parameters.

**3. (35 marks) Page 32 (Problem Set)**

Q:1 Do you believe our capacity for abstract thought is influenced by our language skills? Support your opinion.

Ans: A language is the fundamental tool to abstract thoughts. Without it the thought can not be processed. Much like normal human language, programming languages are the fundamentals of the logical thoughts being abstracted from a human brain. If we consider programming languages, when someone has to solve certain problems which are only solvable if the memory addresses are manipulated. They have no familiarity with the language which is used to manipulate memory addresses like C and C++. It will not be possible for them to solve the problem as the language was the hurdle.

Q:2 What are some features of specific programming languages you know whose rationales are a mystery to you?

Ans: As we learned JavaScript in CS215 there were mysteries solved. Although not everything is clear with this language. It is an interpreted language and being that the main feature is that it can be used for server side scripting as well as client side scripting. But when it comes to error handling on the server side, it does not show any errors on the screen. Although it ‘can’ be shown on the console panel but that's not the most efficient way for many of its use cases.

We also learn PHP in the same class. Personally it is my favorite language when it comes to server side scripting. The flexibility you get with the combination of html and php together is insane. Although every great thing has some bad part in it. When it comes to learning the development with php, language itself is easy but with it you have to learn different frameworks to support your program like Laravel and Symfony.

Q:3 What arguments can you make for the idea of a single language for all program domains?

Ans: If there is only one language the support and community is the best thing we can get out of it. As we all know, learning a new language, its community has the largest effect in it. If there is only one language to do all of it properly, there would be more people learning it and contributing to its community. It will be easier for beginners to focus on their career. It will be easier for companies to hire new programmers as there will be specific needs in a single programming language.

Q:4 What arguments can you make against the idea of a single language for all program domains?

Ans: If there is only one language doing everything, its scope will increase dramatically. As a result it will be costlier to learn. Even if the need is very little a programmer has to learn a lot to solve that problem as it might have a lot of steps to sort out your need in the parts of the language use case. The number of jobs will decrease as there will be more efficient programmers than others as there is only one language. There will be more competition.

Q:9 Explain the different aspects of the cost of a programming language.

Ans: There are many aspects of the cost of a programming language. The costs are either direct or indirect to the programming language. Some these costs are:

1. Training Cost: The cost of teaching programming language in terms of syntax is one step and training a programmer to implement the language in a real life use case is another. The writability of a programming language has a wide impact on this.
2. Writing Cost: The time consumption of writing code in a language is what we call Writing Cost. It also depends mainly on the writability of the language .
3. Compiling and Execution Time: As the name gives out there are a lot of programming languages that face a hard time because of higher compilation and execution cost.
4. Maintenance Cost: How easy it is to manipulate pre existing code can be calculated based on maintenance cost. It also includes system reliability.

Q:10 What are the arguments for writing efficient programs even though hardware is relatively inexpensive?

Ans: Efficient programs makes it easy for the others to debug and manipulate the code. Efficiency describes fastness, clearness and easiness. Inefficient code is always undermined in comparison to efficient code. It is like someone is offering you a 20 megapixel camera in a phone at a price which is good enough but at the same time someone is offering a 400 megapixel camera at double its price which one would you pick?

Q:11Describe some design trade-offs between efficiency and safety in some language you know.

Ans: If we talk about C language, it is mainly used because of efficiency and fastness. Although it is not safe to use. If your code is 100% bug free, we can use C but let's be honest it never is. For example we are assigning a value to an index position ‘i’ in an array a[i] as 20; i is generated in runtime so let say i was assigned -2. C will decrease the memory index -2 and insert the value in some other place which is what we call spraying data in the memory. This is not the case every time as it might give you segmentation faults. Other languages will check that ‘i’ is a valid value and that's why they are slow.

**4. (20 marks) Page 106 (Problem Set),**

Q:12 Describe, in your own words, the concept of orthogonality in programming language design.

Ans: Simplicity of code is what we call Orthogonality of code. If there is a way to do certain tasks in a program, is it the most efficient and simplistic? If this question is answered yes for your program than your program is Orthogonal in compare to the program similar to you but is less simplistic. For example in a website you want to search for something which is indented inside pages. Although you can guide the user to go through the s but the more simpler way is the search bar where users can search their query and get to the page directly.

Q:14 What are the arguments both for and against the idea of a typeless language?

Ans: Let's discuss the arguments in favor of the typeless language first;

* It is easy to write the code and is also more flexible to the general mindset of people. Data types can be assigned to any variable.
* Since it is easy to learn it is easy to write code quickly. The time consumption makes the program to be developed in less time, sparing more time testing and adding new features.
* As data types are no issue any storage location can be used to store values.

Now let's discuss against typeless language;

* Typeless languages are not reliable as there is no type checking.
* Assignments of values and variables are upto developers to check whether they are correct.
* Data corruption possibility increases.

Q:16 What is your opinion of the argument that languages that are too complex are too dangerous to use, and we should therefore keep all languages small and simple?

Ans: Complexity and simplicity is a perspective entity. Nothing can be said complex or simple beside a comparison of better or worse. So everyone has their own perspective of complex programming languages. Personally if making a language more simple and small makes it less efficient than it is a wrong trade off as if you want to have a programming language doing a task efficiently then there will be people learning it no matter how hard and complex it is for one.

Q:18 Languages continually evolve. What sort of restrictions do you think are appropriate for changes in programming language? Compare your answer with the evolution of Fortran.

Ans: There are certain restrictions that must be taken in account when it comes to evolution of a programming language;

* The use of older features must be upgraded rather than adding completely new features.
* Constant addition of new features makes the language complex.
* Removal of useless features which already have a better solution must be removed, like in php we still have mysql keyword which latter used as mysqli.

Fortran has its first version released by the name Fortran 0 which upgraded until Fortran I. Some of the evolution of version I include;

* Fortran I had variable length upgrade up to six from two characters. Input and output formatting and also if statements and loops.
* Fortran II fixed bugs and included addition of independent compilation of subroutines.
* All the way up to Fortran 2003 which supported object oriented programming.

This shows that programming language evolves to be the company standard and beats all the competitors just never realizes it has a negative impact.