

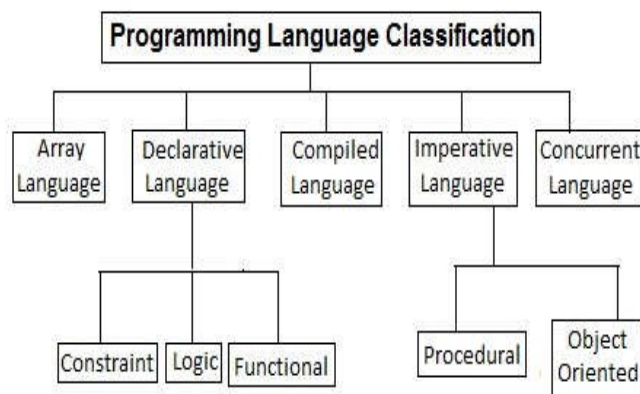


TABLE1: PROGRAMMING LANGUAGES

SN.	LANGUAGE NAME	OVERVIEW	BORN YEAR
[1]	C Language	Imperative Programming	1972
[2]	PROLOG	Logic Programming	1972
[3]	ML	Functional Programming	1973
[4]	Objective C	Object Oriented	1983
[5]	C++	Object Oriented	1983
[6]	Standard ML	Functional Programming	1984
[7]	Erlang	Functional Programming	1987
[8]	Python	Multi-paradigm	1991
[9]	Visual Basic	Event Driven	1991
[10]	Java	Multi-Paradigm	1995
[11]	Ruby	Multi-Paradigm	1995
[12]	Ada 2012	Multi-Paradigm	2012
[13]	Hack	Multi-Paradigm	2014
[14]	Swift	Multi-Paradigm	2014
[15]	C++14	Multi-Paradigm	2014

## II. PROGRAMMING LANGUAGE CLASSIFICATION

Programming languages are classified in to following Categories. Classification of Programming Language shown in Fig. 2.



### A. Array Language

In field of computer technology, array programming languages (which is also called vector languages) aggregate up operations on scalars to apply straightforwardly to vectors, frameworks, and higher-dimensional exhibits. Array programming primitives briefly express far reaching contemplations in regards to data control. The level of concision can be dramatic in some special cases: it is typical to find array programming language one-liners which requires many pages of Java code lines. Some of the array languages are APL, J, FORTRAN and MATLAB.

### B. Declarative Language

Declarative programming is a programming paradigm which tells us about a style of building the structure and components of program that explains the logic of a calculation

without describing flow of control in it. Various languages that uses this style to minimize or remove the side effect by depicting what the program must finish as far as the issue area, rather than telling how to achieve it as an arrangement of the programming dialect primitives (the how being surrendered over to the dialect's execution). This is conversely with basic programming, which executes in explicit steps. Explanatory programming regularly considering program as theory of various formal logic, what's more, calculations as conclusions in that rationale space. Explanatory programming may exceptionally improve forming parallel projects.

1) **Constraint Programming:** Constraint programming is the programming paradigms in which relationship between the variables are expressed in the form of constraints. Constraints differentiate from the normal primitives of basic programming dialects in that they don't demonstrate a stage or grouping of steps to execute, yet rather the properties of a response for be found. This makes constraint programming a type of definitive programming. The constraints utilized as a part of constraints programming are of various sorts: those used as a piece of limitation fulfillment problems (e.g. "X or Y is true" ), those computed by the simplex calculation (e.g. " $m \leq 7$ " ), and others. These constraints are generally inserted with in a programming dialect or given by means of independent software libraries.

2) **Logic Programming:** Artificial Intelligence (AI) is the capacity for a fake machine to act brilliantly. It is a strategy that Computer researchers are using to endeavor to allow machines to reason since it is useful for data representation. In this logic are used to represent knowledge and induction is used to control it. The logic used to represent knowledge in logic writing computer programs is clausal shapes which are the subset of first-order predicate logic. It is utilized on the grounds that first-order logic is surely knew and ready to speak to every single computational issue. Knowledge is controlled utilizing the determination deduction framework which is required for proving theorems in clausal-form logic. The diagram below shows the essence of logic programming. Prolog, PROgramming in LOGic, is a kind of declarative programming paradigms which depends over the thoughts of logic programming, such as those, discussed above. The possibility of Prolog was to make logic look like a programming language and permit it to be controlled by a programmer to advance the research for theorem-proving. All logic programming language can be summed up into a classification of programming languages, called declarative languages. They have a similar thought of analyzing calculations. The calculation is fulfilled by indicating the properties of the right answer. Logic programs can be translated utilizing two primary ideas: truth and logical deduction. Truth infers regardless of whether the calculation in the program is valid, under the usage of the program images. Logical Deduction figures out if a logical clause is an outcome of the program. All logical programming languages have correct operational significance and most bolster the preparing

of limited sets. The projects themselves are guidelines for execution by a compiler. These directions are always interpreted as logical statements and clauses, and the result of the execution is a logical consequence of the calculations in it. In the rationale programming model the software engineer is in charge of indicating the essential sensible connections and does not determine the way in which the induction standards are connected. Thus

Logic + Control = Algorithms

3) **Functional Programming:** Among the various programming languages the functional paradigms is one. It is based on function code which generates the output depending on the provided input values. Here we work on the structures and its components which contain the numeric values for performing the computation besides this here we also categorize the whole logic into sub functions and based on that we perform function calling. Considering an example which contain two input variables  $x$  and  $y$  for which we call the add function to perform the addition of these input variables and as a result we get the following output;  $f(z)=x+y$ .

#### C. Compiled Language

Compiler are not same as interpreter because in interpreter step by step execution of program is performed while in case of compiler every program is translated from the source code into machine code or intermediate code(it is in form of byte code) and after that, interpretation is performed where the intermediate code is executed. By using this language the execution speed becomes faster and code is easily optimized to target hardware. By the name compiler itself it is clear that this language is dependent to the compiler and its working.

#### D. Imperative Language

Imperative programming is a kind of language in which we write the program in the sequence of series which change the state of the computer while in Declarative language program tells "what" to do for achieving the result where as in imperative we perform task to "how" to solve that problem or fulfill it. These programming languages are executed in binary or machine language.

1) **Procedural Language:** Procedural programming is a language, derived from structured programming, based upon the possibility of the procedure call. Procedures, generally calls routines, subroutines, or functions (not to be mistaken for mathematical functions, but instead like those used as a piece of functional programming), basically contain a progression of computational strides to be done. Any given procedure might be called whenever in the midst of a program's execution, including by various procedure or itself. The first major procedural programming languages initially showed up around 1960, including FORTRAN, ALGOL, COBOL and BASIC. Pascal and C were distributed nearer to the 1970s, while Ada was released in 1980. As an example of more modern

procedural language, initially published in 2009. In Procedural programming, Computer processors give hardware support through a stack register and instructions for calling procedures and after processing coming back from them.

2) **Object Oriented Language:** Object-oriented programming structure is a programming model which simplifies software development and maintenance by providing various concept for development of Object, Classification, Generalization and Inheritance, Polymorphism, Data Encapsulation. Any programming language which provides the implementation of above concept is called Object Oriented Programming Language. Different OOP Language may differ in the implementation of OOP's concept. So, C++ and JAVA both of them are Object Oriented Programming Language. JAVA was introduced to facilitate Internet Programming.

### III. SIMULATION RESULT

	Python	SML	C++	Java	Perl	C#	C
<b>Fact(10)</b>	✓	✓	✓	✓	✓	✓	✓
<b>Fact(20)</b>	✓	✓	✓	✓	✓	✗	✓
<b>Fact(30)</b>	✓	✓	✗	✓	✓	✗	✗
<b>Fact(50)</b>	✓	✓	✗	✗	✓	✗	✗
<b>Fact(100)</b>	✓	✓	✗	✗	✓	✗	✗
<b>Fact(250)</b>	✓	✓	✗	✗	✗	✗	✗
<b>Fact(500)</b>	✓	✓	✗	✗	✗	✗	✗
<b>Fact (750)</b>	✓	✓	✗	✗	✗	✗	✗
<b>Fact (999)</b>	✓	✓	✗	✗	✗	✗	✗

In above table we see the comparison done among the various programming language. For comparison we took the program of factorial as it takes large amount of memory for computation as the factorial of  $n$ th number depends on the outcome of factorial of  $(n-1)$ th number (i.e., its previous number computation). We selected the program of factorial for various programming language where, as the outcome we see that the result of the programming language for factorial of large number extend the range and produces the wrong output. We took six programming language out of which only one gives the correct result on providing large input values while another language gave the correct result before 250. Most of them gave output correct till input as 30. This tells us about the dependency of language on memory for computation. Though there is a solution to that by dynamic memory allocation where we use memory at runtime to store the large data but in case of

static memory allocation the strict adhesiveness of language to its syntax bound it to do so.

#### IV. CONCLUSION

In this paper we discussed about the six programming languages we also showed their features and told how much they are good in practical implementation we checked the performance of all the six programming languages where we came to know about the basic problem that they suffer from is that they fail to provide correct result when we exceed their range. We saw that only python is the language that is able to survive the longest while the second language which lasted the long is Perl. However there performance can be improved as nowadays new research in this field help the researcher to know more about the loopholes of these languages and as they are identified the changes are made accordingly so this paper is one step towards making other aware about the performance of some of these languages.

#### REFERENCES

- [1] K. Clark and F. McCabe. Go — a multiparadigm programming language for implementing multi-threaded agents. *Annals of Mathematics and Artificial Intelligence*, 41(2–4):171–206, 2004.
- [2] M. Thielscher. FLUX: A logic programming method for reasoning agents. *Theory and Practice of Logic Programming*, 2005.
- [3] Multi-Paradigm Programming Language". developer.mozilla.org. Retrieved 21 October 2013.
- [4] Frans Coenen (1999-10-11). "Characteristics of declarative programming languages". *Cgi.csc.liv.ac.uk*. Retrieved 2014-02-20.
- [5] Nørmark, Kurt. Overview of the four main programming paradigms. Aalborg University, 9 May 2011. Retrieved 22 September 2012.
- [6] Floyd, R. W. (1979). "The paradigms of programming". *Communications of the ACM*. 22 (8): 455. doi:10.1145/359138.359140
- [7] Peter Van Roy (2009-05-12). "Programming Paradigms for Dummies: What Every Programmer Should Know" (PDF). *Info.ucl.ac.be*. Retrieved 2014-01-27.
- [8] Rajendran, Arunkumar, Nagaraj Balakrishnan, and Mithya Varatharaj. "Malleable Fuzzy Local Median C Means Algorithm for Effective Biomedical Image Segmentation." *Sensing and Imaging* 17.1 (2016): 24..