The Journal of International Lingual, Social and Educational Sciences
Year: 2018, Volume: 4, Number: 1

Submitted: December 2017
Revised: April 2018
Accepted: April 2018

DETERMINATION OF APPROPRIATE PROGRAMMING LANGUAGE BY DECISION SUPPORT SYSTEMS[1]

Muhammer İLKUÇAR[2]

Abstract

With the widespread use of computer and computer systems and the diversity of fields of use, a large number of programming languages with different abilities for different purposes are being developed. So there are many alternates for the programming language. Detailed analysis, experience and expert opinion are needed in order to be able to answer the question of what programming language should be used in one subject. The programming languages to be determined according to the purpose of use in the study are aimed to help decision making by taking expert opinions, evaluating language characteristics and certain criteria in the market (popularity, job opportunities, coding convenience, etc.) and sorting with analytical hierarchical programming. Considering criteria such as user interface, simple of coding, popularity, job-finding facilities, and lots of finding documents from popular programming languages designed to improve the application of new and less-experienced programmers a ranking was made by analytical hierarchical programming with pairwise comparisons according to expert opinions. As a result of ranking, Java is the first with 28%, second is C# with 26% and C ++, C, VB.NET is listed as 16%, 15% and 15% respectively.

Key Words: *AHP, Programming Language, Multi Criteria Decision.*

[1] This paper oral presentation at 1st International Symposium on Social and Educational Sciences Research (ISCER 2017), October 3-5 Antalya/Turkey.

[2] Asistant Prof. Dr., Mehmet Akif Ersoy University, imuammer@mehmetakif.edu.tr

INTRODUCTION

The decline in computer prices has made it easier to access. This has accelerated the development of computer technology. Its models and features have increased, and it has been using widespread. Parallel to these developments, computers are used for many different purposes. This has led to the development and diversification of programming languages. There are still millions of applications, hundreds of programming languages, and new programming languages that are currently serving different purposes. Each programming language offers different advantages and disadvantages depending on where it is used. Therefore, choosing a programming language according to the job to be done will give the programmer many convenience or many difficulties. Programming languages can be divided into different categories according to their abilities. In Figure 1, a classification is made by taking into consideration the skills of programming languages frequently used in the market. This work to be done according to the figure is divided into four main categories as application development, data processing, website making and mobile application development. In addition; programming language is possible to distinguish between different categories such as graphics processing, mathematical calculation, simulation, fast processing, compilation methods, close to machine language, web [3]environment, small smart device programming, etc.

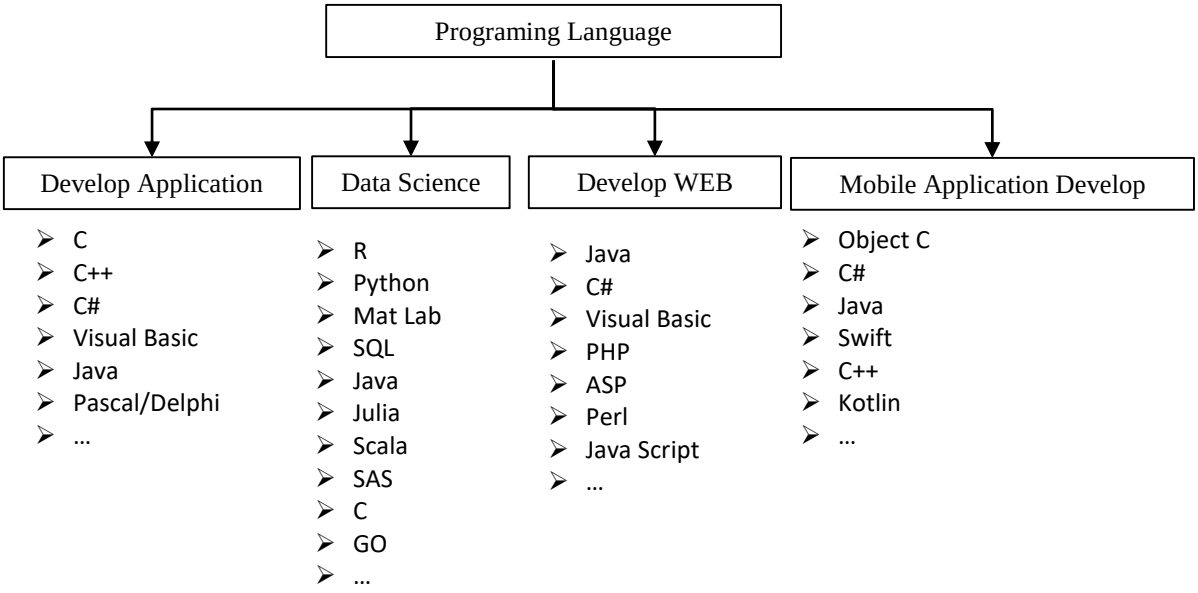

Figure 1. Classification of programming language according to its skills

Comparisons are made according to different characteristics of programming languages in the literature. TOBE (2017) makes popularity ranking by comparing the language of about 100 programs with different criteria considering the data of the previous year every year. Basically, it takes into account information such as the effective use of the language on engineering at the world scale, courses and third party producers. DataCamp (2017) uses the R and Python programming languages with each other, graphics capabilities, IDEs, usability, and etc. have made a detailed comparison. Burtch Works (2017) compared the usage rates of R, Python and SAS for different disciplines. Chen (2010) compares popular programming languages such as C, C++, C#, and Java with each other by making specifications and benchmark tests. Partibha and Khokhar (2015) a comparison was made comparing the advantages and disadvantages of C, C++, C# and Java languages in terms of their data type, structure, array, class structure, compiler technology. AnalatiscVidhya (2017) made a comparison comparing R, Python, and SAS, taking into account factors such as availability/price, ease of use, data processing capacity, graphics capacity, development tools, job opportunities and scored them (1 lowest, 5 highest).

In this study is intended to help programmer candidates to choose a programming language that is not clear about what programming language will be used in a particular case. By using the five programming languages differences by specific skills from popular programming languages. It will help to make a ranking and decision by using Analytical Hierarchical Programming (AHP) method in the direction of expert opinions.

ANALYTIC HIERARCHY PROGRAMMING

The AHP is a decision support method that uses the non-numerical data presented by Saaty (1980) to perform pairwise comparisons according to criteria among multiple options. Figure 2 shows the hierarchical structure of a general-purpose problem. The aim here is; alternatives are compared with each other according to the criteria, alternatives are sorted and support is decision maker. First of all, criterions are compared with each other and importance weights are determined according to each other. Then each alternative *i*. must be compared with each other according to the criterion.

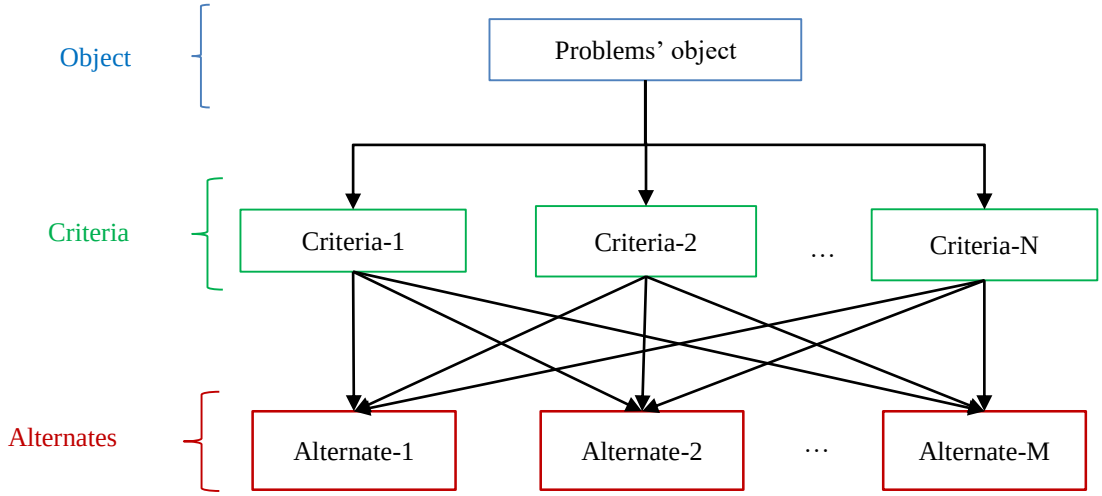

Figure 2. Hierarchical structure of AHP problems

The scale proposed by Saaty (1980) is used for pairwise comparison in AHP. Accordingly, the relative judgments of the criteria are digitized with a number between 1 and 9, as shown in Table 1. The numbers 1, 3, 5, 7, 9 are the final judgments, and 2, 4, 6, 8 are the interim values (compromise values) between the two judges.

Table 1. The Score of Comparison of the Relative Judgments of the Criteria (Saaty,2008)

Score	Judgements
1	***Equal Importance)***
2	Weak or Slight Importance
3	***Moderate Importance***
4	Moderate Plus Importance
5	***Strong Importance***
6	*Strong Plus Importance*
7	***Very Strong Importance***
8	Very Very Strong Importance
9	***Extremely Importance***

In pairwise comparisons, the weight criterion should be symmetric with $a_{i-j} = 1/a_{j-i}$ when the criterion *i* and criterion *j* are compared. A criterion is itself $a_{i-j} = a_{j-i} = 1$. In order not to make an error, only the a_{i-j} comparison should be performed and the a_{j-i} comparison should be calculated as $a_{i-j} = 1/a_{j-i}$. This way the sample is filled in accordance with the judgments in Table 1. As seen in Table 2, a matrix of comparison is created. From the obtained comparison matrix (Table 2), the priority weight values between the options are calculated by using any of the eigenvector, geometric mean or arithmetic average methods. According to time (1980), for the eigenvector *w* of matrix A, it is written as $AW = \lambda_{max}W$. Where λ_{max} is the maximum value of the normalized eigenvector of the matrix A.

Table 2.Four criteria comparison matrix example

	i-j	A_1	A_2	A_3	A_4	*Priority Wight*
	A_1	1/1	5	1/2	1	0,29
$A=$	A_2	1/5	1/1	3	1/4	0,14
	A_3	2	1/3	1/1	1/2	0,17
	A_4	1	4	2	1/1	0,38

The consistency of the data in the comparison matrix should be investigated. Consistency Ratio (CR), Consistency Index (CI), and Random Inconsistency Index (RI) information are needed for this data. For AHP results to be reliable, the consistency rate should be less than 10% (CR<0.1) (Dozic and Kalic, 2014). The RI value for the problems with small number of variables (n = 10) is given in Table 3 (Saaty, 1987). The consistency ratio (CR) of the comparison matrix is calculated by the following equations as:

$$CI = \frac{\lambda max - n}{n - 1}, \qquad CR = \frac{CI}{RI} \qquad \text{n: problems' variable count}$$

Table 3. Random inconsistency index

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0,58	0,90	1,12	1,24	1,32	1,41	1,45	1,49

SAMPLE

This study has been done to answer the question "which programming language proper for me?" For prospective software developers who are at the beginning level or have little knowledge about programming. C, C++, C#, Visual Basic.NET, etc., taking into account such situations as programming languages, expert opinions, popularity at world scale, ease of coding, easy access, local language (for instance: Turkish) resource finding, future promise, programming languages have been defined. For these programming languages, criteria such as programming interface (GUI), document retrieval, coding convenience, popularity, job opportunities will be used considering the situation of users. According to these criteria, the experts will be able to make pairwise comparisons with each other in order to sort out the alternatives with the help of AHP and help the user to decide. The hierarchical structure of the study in Figure 3 is given. The object here is "Which programming language is best or proper for me?" And the alternatives are "C, C++, C#, VB.NET and Java". The criteria are "user friendly IDE, access abundant document, easy coding syntax, popularity and job opportunities". According to this, alternatives will be listed using AHP.

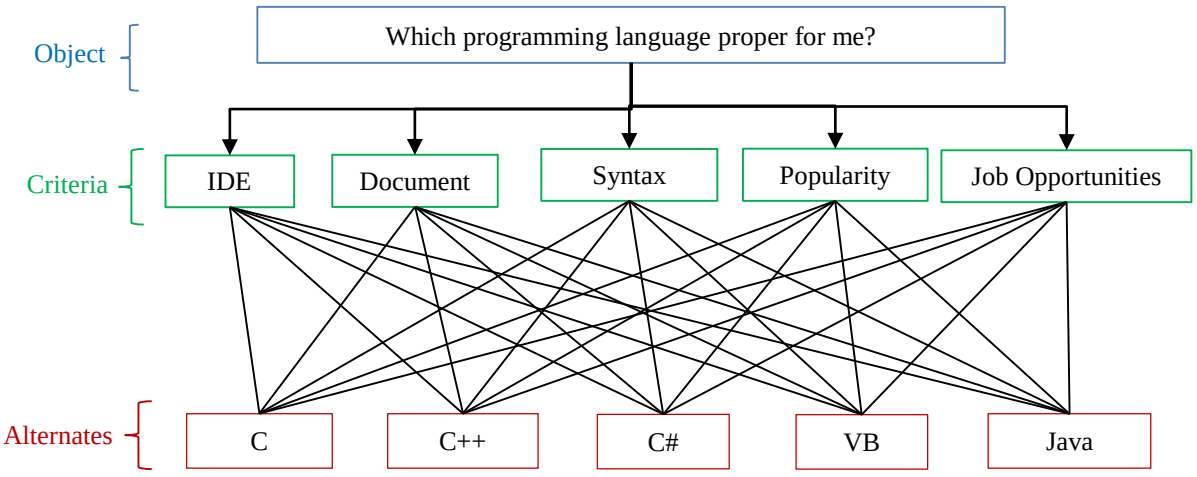

Figure 3.Object, criteria and alternates hierarchical structure

First of all, criteria should be compared with each other and importance weight should be determined. In Table 4, the matrices of comparison of the criteria are given in the direction of expert opinion. As the comparison values, the arithmetic mean of the three expert opinions was tabulated as a single data. As a result of comparison, the consistency ratio (CR) of the scorers was calculated as 0.089. The comparison data is 0.089<0.10 then consistent. If the consistency rate is greater than 10%, it will mean that there is inconsistency in comparisons. In this study we make an application program, experts were provided to enter the comparison scores via the program. Thus, it is possible to see whether the comparison is instantaneous and therefore it is possible to make a healthier comparison. In order to determine the weight of the criteria according to Table 4, the geometric mean method of the rows is used. The line geometric mean of each criterion is calculated for this. These values are normalized so that each criterion has priority order (weight values) relative to each other. According to Table 4; job opportunities with 0.544 are the greatest value of weighting criteria, the lowest weighting value with 0.069 has been the GUI.

Table 4. Matrix of comparison of criteria

	GUI	*Document*	*Coding*	*Popularity*	*Job Facility*	*Geometric Mean*	*Priority Vector (PV)*
GUI	1,00	0,33	0,25	2,00	0,20	0,506	0,069
Document	3,00	1,00	0,33	3,00	0,20	0,902	0,123
Coding	4,00	3,00	1,00	4,00	0,20	1,572	0,215
Popularity	0,50	0,33	0,25	1,00	0,13	0,349	0,047
Job Facility	5,00	5,00	5,00	8,00	1,00	3,981	0,544
Column Sum	13,5	9,666	6,833	18	1,725	7,310	
λ_{max} *(Sum*PV)*	0,935	1,193	1,469	0,860	0,939	=5,397	
λ_{max} =	5,397						
$CI=(\lambda_{max}-n)/(n-1)$	0,099						
CR=CI/RI	0,089						

Once the weight values of the criteria have been determined, the alternatives should be compared with each other according to each criterion. Since there are five criteria in the study, alternatives are compared with these five criteria one by one. Comparison matrices and consistency ratios of the alternatives between Tables 5-9 are given. When comparison scores are given, care should be taken to ensure that consistency rates are less than 10%. The pairwise comparison values in Tables 8 and 9 are determined to taking into account the data of Itjobswatch (2017) and TOIBE Index (2017).

Table 5. Comparison matrix of alternatives by GUI criteria

	C	*C++*	*C#*	*VB*	*Java*	*Geometric Mean*	*PV*
C	1,00	1,00	0,14	0,14	0,20	0,332	0,044
C++	1,00	1,00	0,14	0,14	0,20	0,332	0,044
C#	7,00	7,00	1,00	1,00	4,00	2,873	0,382
VB	7,00	7,00	1,00	1,00	4,00	2,873	0,382
Java	5,00	5,00	0,25	0,25	1,00	1,093	0,145
Column Sum	21	21	2,535	2,535	9,4	=7,506	
λ_{max} *(Sum*PV)*	0,931	0,931	0,970	0,970	1,369	=5,172	
λ_{max} =	5,172						
$CI=(\lambda_{max}-n)/(n-1)$	0,043						
CR=CI/RI	0,038						

Table 6. Comparison matrix of alternatives by document criteria

	C	*C++*	*C#*	*VB*	*Java*	*Geometric Mean*	*PV*
C	1,00	5,00	0,50	0,50	2,00	0,332	0,044
C++	0,20	1,00	0,33	0,50	2,00	0,332	0,044
C#	2,00	3,00	1,00	2,00	2,00	2,873	0,382
VB	2,00	2,00	0,50	1,00	2,00	2,873	0,382
Java	0,50	0,50	0,50	0,50	1,00	1,093	0,145
Column Sum	5,7	11,5	2,833	4,5	9	5,563	
λ_{max} *(Sum*PV)*	1,230	1,202	0,961	1,066	0,928	5,389	
λ_{max} =	5,389						
$CI=(\lambda_{max}-n)/(n-1)$	0,0974						
CR=CI/RI	0,087						

Table 7. Comparison matrix of alternatives by coding criteria

	C	*C++*	*C#*	*VB*	*Java*	*Geometric Mean*	*PV*
C	1,00	2,00	0,50	0,50	5,00	1,201	0,217
C++	0,50	1,00	0,50	0,50	4,00	0,870	0,157
C#	2,00	2,00	1,00	1,00	2,00	1,515	0,274
VB	2,00	2,00	1,00	1,00	2,00	1,515	0,274
Java	0,20	0,25	0,50	0,50	1,00	0,416	0,075
Column Sum	5,7	7,25	3,5	3,5	14	5,519	
λ_{max} *(Sum*PV)*	1,240	1,1435	0,961	0,961	1,055	=5,362	
λ_{max} =	5,362						
$CI=(\lambda_{max}-n)/(n-1)$	0,090						
CR=CI/RI	0,081						

Table 8. Comparison matrix of alternatives by popularity criteria

	C	*C++*	*C#*	*VB*	*Java*	*Geometric Mean*	*PV*
C	1,00	1,00	2,00	4,00	0,50	1,319	0,217
C++	1,00	1,00	2,00	2,00	0,33	1,059	0,174
C#	0,50	0,50	1,00	2,00	0,25	0,659	0,108
VB	0,25	0,50	0,50	1,00	0,20	0,416	0,068
Java	2,00	3,00	4,00	5,00	1,00	2,605	0,429
Column Sum	4,75	6	9,5	14	2,28	6,063	
λ_{max} *(Sum*PV)*	1,034	1,048	1,034	0,961	0,981	=5,060	
λ_{max} =	5,063						
$CI=(\lambda_{max}-n)/(n-1)$	0,015						
CR=CI/RI	0,013						

Table 9. Comparison matrix of alternatives by job facility criteria

	C	*C++*	*C#*	*VB*	*Java*	*Geometric Mean*	*PV*
C	1,00	0,33	0,50	7,00	0,33	0,827	0,132
C++	3,00	1,00	0,50	5,00	0,33	1,201	0,192
C#	2,00	2,00	1,00	5,00	1,00	1,820	0,291
VB	0,14	0,20	0,20	1,00	0,20	0,257	0,041
Java	3,00	3,00	1,00	5,00	1,00	2,141	0,342
Column Sum	9,142	6,533	3,2	23	2,867	6,248	
λ_{max} *(Sum*PV)*	1,211	1,255	0,932	0,949	0,982	=5,331	
λ_{max} =	5,331						
$CI=(\lambda_{max}-n)/(n-1)$	0,082						
CR=CI/RI	0,074						

Table 10 summarizes the data from the criteria and alternatives. This table (Table 4) and the results obtained from Table 5-9 where the alternatives are compared according to the criteria. According to the criteria from Table 5-9, the weight values from Table 4 are multiplied by the data and the row (alternatives) sum of the products gives the priority order (Table 11). This table shows the order of the alternatives. The alternative ordering according to Table 11 is as follows: Java with 28% and C# with 26%. C++, C, and VB.Net is around 15%. The user can choose the Java or C# programming language accordingly.

Table 10. The pairwise comparison weight obtained from criteria and alternate

	GUI (0,069)	Document (0,123)	Coding (0,215)	Popularity (0,047)	Job Facility (0,54)
C	0,044	0,215	0,217	0,217	0,132
C++	0,044	0,104	0,157	0,174	0,192
C#	0,382	0,339	0,157	0,108	0,291
VB	0,382	0,237	0,274	0,068	0,041
Java	0,145	0,103	0,274	0,429	0,342

Table 11. Alternate priority order

	GUI (0,069)	Document (0,123)	Coding (0,215)	Popularity (0,047)	Job Facility (0,54)	Priority
C	0,003	0,026	0,046	0,010	0,072	0,157
C++	0,003	0,012	0,033	0,008	0,104	0,160
C#	0,026	0,041	0,033	0,005	0,158	0,263
VB	0,026	0,029	0,059	0,003	0,041	0,158
Java	0,010	0,012	0,059	0,020	0,186	**0,287**

CONCLUSION

In this study, an answer was sought in order to find out which programming language should be used by beginner and lesser-known programmer candidates who want to develop applications. Accordingly, five of the application development programming languages popular in the market were selected and sorted by AHP according to criteria such as user interface, ease of coding, easy resource finding, popularity and job opportunities. This will help you to decide. In the study, expert opinions and language ratings were taken into consideration in order to compare the criteria against each other and to compare the languages according to the criteria. It is suggested that someone who will learn the new programming

language to develop an application according to the work and find a job to start with Java C# or Java programming language.

The work can be taken on the web and provided on-line. The system can be made customizable so everyone can help them decide their own criteria and their own alternatives and decide by scoring according to their own judgments. The program can be developed to provide decision support services for different matches.

REFERENCES

AnalaticsVidhya (2017). *https://www.analyticsvidhya.com/blog/2014/03/sas-vs-vs-python-tool-learn/* (Access date: 03.03.2017)

Burtch Works (2017). *http://www.burtchworks.com/2016/07/13/sas-r-python-survey-2016-tool-analytics-pros-prefer/* (Access date: 03.03.2017)

Chen H. (2010). *Comparative Study of C, C++, C# and Java Programming Languages, Vaasan Ammattikorkeakoulu University of Applied Sciences Degree Program of Information Technology*, (2010, Thesis)

DataCamp (2017). *https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis#gs.uC28Gs8*, (Access date: 03.03.2017)

ITJobWatch, (2017). *https://www.itjobswatch.co.uk/IT-Job-Market/UK/Programming-Languages*, (Access date: 09/11/2017)

Dozic S. D. and Kalic M., (2014). *An AHP approach to aircraft selection process*. *Transportation research Procedia* 3(2014) 165-174.

Partibha Mrs., Khokhar Mrs. A. (2015). *Comparative Study of C, C++, C# and Java Programming Languages, International Journal of Enhanced Research in Management & Computer Applications, ISSN: 2319-7471 Vol. 4 Issue 6, June-2015, pp: (7-12), Impact Factor: 1.296, Available online at: www.erpublications.com*.

Saaty L. T., (2008). *Decision making with the analytic hierarchy process*, Int. J. Services Sciences, Vol. 1, No. 1, 2008.

Saaty L. T., (1990). *The Analytic Hierarchy Process In Conflict Management*, I*nternational Journal of Conflict Management, Vol. 1 Issue: 1, pp.47-68*, https://doi.org/10.1108/ eb022672

Saaty L. T., (1987). *The Analytic Hierarchy Process-What It Is and How It Is Used. Math/ Modelling*, Vol. 9, No. 3-5, pp. 161-176, 1987.

TIOBE Index (2017). *http://www.tiobe.com/tiobe-index/* (Access date: 09.11.2017)