[1,2]Róbert PAP-SZIGETI, [2]Attila PÁSZTOR

AN EXAMINATION OF SKILLS AFFECTING THE EFFECTIVENESS OF PROGRAMMING

[1]Bolyai János Secondary Grammar School, 49 Irinyi str. Kecskemét, HUNGARY
[2]Department of Informatics, GAMF Faculty of Engineering and Computer Science, John von Neumann University, 10 Izsáki str. Kecskemét, HUNGARY

Abstract: In this article we examined the effect of the development of reading comprehension and word recognition on the performance of understanding algorithms and programming tasks. Additionally, we examined the role of algorithmization skills, knowledge of programming language elements and the use of the IDE in the successful solution of advanced graduation programming tasks. Based on the results of our surveys taken with five subtests on the sample of high school and IT engineering students (n = 128), we present how these skills are layered on each other.
Keywords: Teaching of programming, Algorithms, Reading comprehension

1. INTRODUCTION

For a wide range of students, learning basic programming skills, understanding and applying algorithms are essential for effective application of modern informatics equipment. Although small groups of users of informatics equipment engage in advanced coding, remembering and adapting complex algorithms are frequently required for effective use of applications.

At the end of the previous century, numerous studies dealt with connections between programming skills and general thinking abilities (e.g. *Kurland* et al., 1981; *Lehrer* et al., 1988; *Pea* and Kurland, 1984). In the last two decades, studies have been directed to identify those components, which constitute knowledge, ability and motivation system of programming (e.g. *Lahtinen* et al., 2005; *Milne* and *Rowe*, 2002; *Robins* et al., 2003).

Students' motivation connected to programming was examined in previous studies of the authors of this article. We examined the impact of the application of different teaching methods on self-concept and skills of programming (*Pásztor* et al., 2010).

In this article, we describe the connection between secondary school and BSc students' programming performance and some prerequisites of programming. The fields of our examination were word recognition, reading comprehension, knowledge of syntactical elements of a formal programming language and the usage of the IDE (integrated development environment).

2. RESEARCH QUESTIONS

The two basic research questions are (1) whether reading comprehension and its component, the word recognition are prerequisites for algorithmization required at the level of *A plus* level exam[1]. (2) Can it be stated that skills and knowledge required at the level of *A plus* level exam can be classified into four, filter-like components based on each other. Features of the filters have two basic criteria: a threshold level of the lower level components is required for the operation of the higher level skills. However, the particular differences in knowledge beyond the threshold level do not significantly influence the particular differences measured at a higher level. Since valuation of details are not carried out on the scale the threshold levels are not standardized. Thus nominal values of the threshold level of a particular variable may differ.

3. METHODS

■ Sample

The research sample (n = 128) consisted of high school students and BSc undergraduates who study algorithmization and coding as an integral part of their curriculum. The participants of the high school subsample were selected from the students studying informatics and students preparing for their A plus level exam. Other participants were IT engineering undergraduates. The composition of the sample is shown in *Table 1.*

This subtest was not taken by the 9th and 10th grade students since not all the necessary elements of the programming test (see next chapter) were included in the 9th grade curriculum. The BSc IT engineers' subsample was narrowed to 29 students due to organisational reasons; therefore, the results and correlation

[1] An advanced level exam taken by students in order to get admitted to university

of the programming task are analysed based on the data of 80 people. The differences between the results of the subsample compared to the one of the whole sample are taken consideration.

Table 1. Division of the sample according to subsamples and grades

	high school subsamples				BSc IT engineer
	grade 9	grade 10	grade 11	grade 12	
whole sample (*n*)	22	21	23	28	34
sample of programming subtest (*n*)	0	0	23	28	29

The composition of the sample, the proportion of the subsamples significantly differ from pre-test (*Pap-Szigeti*, 2018), thus their results will not be compared.

■ **Instruments and procedures**

The test battery consists of a brief questionnaire and five subtests. The questionnaire examines classifying variables and components of self-concept connected to programming skills and reading comprehension. The subtest measuring skills of word recognition consisted of the exercises of the nation-wide competency test (86 items) and exercises related to IT vocabulary (18 items) (Cronbach-α = 0.89). The rest of the subtest was developed by the authors. The subtest measuring reading comprehension (16 items; Cronbach-α = 0.71) included tasks which required simple information retrieval based on the text, inferential reading comprehension and reflection on the text. The lower level of reliability might be the consequence of the ceiling effect ($\bar{x}$ = 81 %points), distribution skewed left might direct us to the same conclusion. (Kolmogorov-Smirnov z = 2.139; $p < 0.001$).

The other four subtests (algorithmization, elements of programming language, the usage of the IDE, programming) were conducted in three instead of four parts. Since the authors had to monitor the time frame and the results of the small sample warranted the three subtests. The algorithmization subtest incorporated 32 items on the three applications levels (Cronbach-α = 0.92). 30 items of programming language and the 22 items related to integrated development environment tasks were put into the same subtest (Cronbach-α = 0.90). The programming task was similar to the *A* level exam which included 30 items (Cronbach-α = 0.91). The algorithmization (Kolmogorov-Smirnov z = 1.42; p = 0.026) and programming subtests (z = 1.551; p = 0.040) revealed bimodal distribution. The distribution of the subtest of the language elements and IDE can be considered as normal.

The examinations took place in spring 2018. The pilot test was conducted at the same time as the large sample test (*Pap-Szigeti*, 2018). The data of the pretest were separately analysed from the large sample. The surveys were conducted digitally in the test system created by the first author (*Török* et al., 2016; *Török* and *Pap-Szigeti*, 2018). 92% of the questions of the survey and the test items of the first four subtests were corrected automatically (single or multiple choice test, grouping into sets). The rest of the items were manually corrected by the authors according to the key. The correction of the programming tasks was done by hand.

4. RESULTS

■ **Results of word recognition and reading comprehension**

The average level of word recognition is equal to ones achieved at the nation-wide survey since the whole sample $\bar{x}$ = 87 %points, s = 6.4 %points. Word recognition and IT vocabulary revealed a medium correlation (r = 0.316; $p < 0.001$). Everybody reached 60 %points in word recognition, yet, about one third of the sample (40 people) reached maximum 50 %points. There was no difference in the level of understanding everyday words between grades ($\bar{x}$ = 87.7...93.8; ANOVA F = 2.140; p = 0.138). However, when it comes to the IT vocabulary, 9th grade students' and undergraduates' output was 12.5% lower compared to the grades between 10 and 12, which is significant. (ANOVA F = 9.379; $p < 0.001$). A possible reason for this difference can be that the 9th graders and undergraduates studied programming for a shorter time.

The general results of the reading comprehension $\bar{x}$ = 82 %points (s = 12.5 %points) one eighth of the sample (16 people) achieved less than 66 %points. There is no significant difference between the results of the particular grades (ANOVA F = 1.114; p = 0.096). And yet, there is a medium correlation between reading IT vocabulary and results of the reading comprehension (r = 0.540; $p < 0.001$. Thus, individual discrepancies of the reading comprehension go hand in hand with the differences of the IT vocabulary. Those students who performed under 50% in word recognition of the IT vocabulary achieved 15% less in reading comprehension (F = 2.338; p = 0.129; t = 5.908; $p < 0.001$).

■ **Results of the algorithmization and the correlation with the components of reading skills**

The average results of algorithm partial tasks are $\bar{x}$ = 43 %points (s = 24.5 %points). The distribution of the performance is bimodal. The reason for the bimodality is not those, who started programming recently – 9th grade students and undergraduates – performed significantly lower (beginners: $\bar{x}$ = 41.2 %points; rest $\bar{x}$ = 44.7

%points). There is a compelling evidence, that there is a correlation between the results of the algorithmization subtest and the length of the studies spent with programming (r = 0.483; p < 0.001). The average achievement of the participants having been learning programming for 1-2 years is 22.6 %points; however, the ones who have been programming for 2-5 years achieved 49%.

Those who performed less than 50% in word recognition of IT vocabulary were able to solve the algorithm subtest with 20% less average points, which is a definite difference, compared to the ones who achieved above 50% of word recognition ($\bar{x}$ = 29.4 %points; s = 20.9 %points; $\bar{x}$ = 49.5 %points; s = 23.5 %points). The results of those students who performed more than 50% of word recognition of IT vocabulary (n = 88) do not show any correlation between the efficiency of the algorithmization and word recognition of IT vocabulary (r = 0.154; p = 0.155). This means that IT vocabulary influences the efficiency of the algorithmization as a filter.

The difference between the two subsamples above appeared at more difficult application levels (*Table* 2). At the level of interpretation, the standard deviations of both subsamples are great as expected, in reading IT vocabulary, the individual discrepancies are greater.

Table 2. Average (std. dev.) of the algorithmization according to the application levels and skills of word recognition IT vocabulary

	recognition and remembrance	execution, decoding	recognition and remembrance
number of items	12	8	12
IT vocabulary word recognition <= 50 %points	48.1 %p (27.4 %p)	25.3 %p (24.4 %p)	28.6 %p (22.6 %p)
IT vocabulary word recognition >= 50 %points	54.3 %p (30.3 %p)	53.1 %p (27.5 %p)	42.7 %p (26.3 %p)
difference	F = 1.15; p = 0.285 t = 1.10; p = 0.273	F = 1.77; p = 0.186 t = 5.33; p < 0.001	F = 6.63; p = 0.011 d = 3.01; p < 0.001

The insufficient reading comprehension skills cause obvious differences. People scoring less than 65% in reading comprehension achieved significantly less in the algorithmization subtest compared to other test subjects ($\bar{x}$ = 18.7 %points; s = 16.4 %points; $\bar{x}$ = 45.8 %points; s = 23.8 %points; t = 3.84; p < 0.001). The difference appeared at a higher application level as well. If students achieved more than 65% in reading comprehension, there is no correlation between the reading comprehension scores and the results in algorithmization (r = 0.167; p = 0.108). Due to this, reading comprehension is a filter like condition of algorithmization.

■ **Results of subtest of the language elements and integrated development environment**

The combined average of the two subtests is $\bar{x}$ = 52.7 %points, the standard deviation is s = 24.3 %points; language elements: $\bar{x}$ = 54.0 %points; IDE: $\bar{x}$ = 49.5 %points; the distribution of results is normal. The correlation between the two partial elements i.e. language elements and results of the integrated development environment is significant (r = 0.703; p < 0.001). Thus, analysis will be carried out with the results of the combined results of the two elements. Since correlation was high the hypothesis that any of the components may have a filter effect on each other must be rejected.

The results of the contracted variables clearly reveal a medium correlation with the time span the subject has learnt (r = 0.449; p < 0.001) programming. The subjects having learnt it for 1-2 years achieved average 26.7 %points, the average of ones having learnt it for 3-5 years has 58.2 %points on average.

There is a medium correlation between word recognition of IT vocabulary and the combined average of the two subtests (r = 0.426; p < 0.001). There is a significant difference between the averages of the two subsamples if the line is drawn at the 50% of IT vocabulary (F = 3.751; p = 0.056; t = 2.200; p = 0.030). Neither knowledge of language elements nor the usage of the IDE shows any correlations with reading comprehension (r = 0.175; r = 0.160; n.s.). This is a clear sign of that the subtest encloses syntactical elements on the one hand and technical ones on the other. None of these require the higher level of reading comprehension. There were algorithmization tasks, which required execution (r = 0.423; p < 0.001) and interpretation (r = 0.618; p < 0.001) in the algorithmization subtest. This shows a medium or high correlation with the results of the elements of the programming language subtest. It can be assumed that the relation is not that causal but cooperative. Those subjects, who achieved good results might be better at problem solving. This is partially corroborated by the fact that these correlation constants decreased if the effect of one particular variable is controlled by partial correlation. This variable is the time span of years spent in programming (r = 0.383; r = 0.558; p < 0.001).

The subtests of the language elements and IDE reveal results as expected when the application level of the tasks is taken into account. Achievements are slightly decreasing when reaching higher levels (recognition, remembrance: $\bar{x}$ = 55.5 %point; s = 20.6 %points; execution, decoding $\bar{x}$ = 52.4 %points; s = 23.2 %points; interpretation: $\bar{x}$ = 44.6 %points; s = 21.6 %points). The level of interpretation is significantly lower than the levels of execution and decoding (t = 5.38; $p < 0.001$).

■ **Results achieved in the programming subtest**

As it was mentioned the programming subtest was completed by 80 participants. The results of this subsample in the first four subtests can be seen in *Table 3*. The average of the reduced sample is significantly higher at every subtest compared to those who did not complete the programming subtest. This phenomenon can be explained by the fact that students who did not complete the programming subtest scored lower in reading comprehension.

Table 3. Average (std. dev.) results of the programming subtest in the first four subtests

word recognition	reading comprehension	algorithmization	language elements and usage of the IDE
89.8 %p (4.1 %p)	84.6 %p (10.6 %p)	53.2 %p (22.3 %p)	59.1 %p (20.3 %p)

The most important statements written in previous three chapters can be uphold in the subtest who wrote the subtest of programming:

≡ Those who achieved less than 50% at word recognition of IT vocabulary achieved significantly lower results in reading comprehension compared to achievers over 50% ($\bar{x}$ = 79.2 %points; $\bar{x}$ = 88.3 %points; F = 0.006; p = 0.937; t = 3.172; p = 0.002).

≡ Results of the algorithmization subtest significantly correlate with the years spent with programming (r = 0.501; $p < 0.001$). The average of the participants spent 1-2 years with programming is 34 %points; however, ones spent 3-5 years with programming have 55 %points.

≡ Those who achieved less than 50% in word recognition of IT vocabulary achieved 11% less in the algorithmization subtest compared to those who performed well in reading IT vocabulary ($\bar{x}$ = 44.1 %points; s = 21.7 %points; $\bar{x}$ = 55.2 %points; s = 22.0 %points; F = 0.034; p = 0.854; t = 2.132; p = 0.036). The difference appeared in the level of execution, decoding first, (difference of the average of the two subsamples is 18.3%.

≡ IT vocabulary operates as a filter. There is no correlation between efficiency of algorithmization and the skill of reading IT vocabulary (r = 0.147; p = 0.241) in those participants' performance who exceeded 50% in reading IT vocabulary.

≡ Those, who achieved less than 65% in reading comprehension had reached 15.7% lower average in the algorithmization test, which is significant compared to ones who did better at reading comprehension. Reading comprehension can be seen as a filter: there is no significant correlation between the results of algorithmization and reading comprehension for those who achieved above 65% in reading comprehension (r = 0.121; p = 0.296).

≡ There is a weak to medium correlation between language elements, usage of integrated development environment and reading IT vocabulary (r = 0.314; p = 0.005), but there is no correlation.

≡ There is a weak to medium correlation between the results of the language elements subtest of programming and execution, interpretation, which are components of the algorithmization subtest.

The average test results of programming is $\bar{x}$ = 72.1 %points. The distribution is bimodal (Kolmogorov-Smirnov z = 1.551; p = 0.040). There is no significant difference between the grades (ANOVA F = 2.079; p = 0.167). The averages vary between 70.1% and 74.3%. There is no significant correlation between the results of the programming subtest and the time span of years spent in programming (r = 0.218; p = 0.053).

The results of the first four subtests and the results of programming subtest suggest that there is a strong correlation with the IDE subtest (r = 0.612; $p < 0.001$), and weak-medium correlation with the interpretation level of the algorithm subtest (r = 0.292; p = 0.011). Let us consider all the items of the first four subtests as an independent variable, then let us perform a linear regression analysis on those variables. According to the results of the aforementioned analysis 93.7% of variance of programming subtest can be explained by the variance of items (R^2 = 0.937; F = 10.565; $p < 0.001$). Thus, the four subtests together can predict individual differences of the programming subtest.

■ **Relationship among algorithmization, programming language elements, IDE and programming subtests**

The data of the just 80 people completed the programming subtests are examined in this chapter.

In the subsample, the level of 60 %points achieved in the algorithmization subtest operates as a filter when it comes to the subtests of programming language elements and IDE (n_1 = 40; $\bar{x}$ = 52.5 %points; s = 19.3 %points) (F = 0.032; p = 0.859; t = 3.046; p = 0.003). There is a significantly weaker output at the test of elements of programming language and IDE compared to those whose output was above 60% in the algorithmization subtest (n_2 = 40; $\bar{x}$ = 65.7 %points; s = 19.4 %points). At the same time in case of participants achieving more than 60%, the individual differences of algorithmization do not significantly influence the individual differences of elements of programming language and IDE (r = 0.253; p = 0.083).

The filter function cannot be proved between the components of the programming language subtest and IDE subtest. The correlation coefficient is higher for the two components than those who solved the programming subtest (r = 0.771; $p < 0.001$).

The consequence of the high correlation (r = 0.612; $p < 0.001$) is that there is no filter function in the results of the elements of programming language and IDE, and the programming subtests. Results indicate that development of the three areas move together.

Algorithmization has a filter like effect on programming; 40 people, who achieved less than 60% at the algorithmization subtest, have significantly lower output at the programming subtest compared to the other 40 people whose output was above 60% ($\bar{x}$ = 64.6 %points; s = 25.3; $\bar{x}$ = 76.6 %points; s = 22.4; F = 2.503; p = 0.118; t = 2.952; p = 0.007). Yet, in case of the output, above 60% on algorithmization there is no correlation with the results of the programming subtest (r = 0.157; p = 0.332).

5. CONCLUSIONS

The individual differences in word recognition due to the high average do not explain differences in reading comprehension, however, word recognition of IT vocabulary is strongly correlated to reading comprehension. It can be supposed that the effect of general learning skills and motivation can be recognised.

There is a filter function of word recognition of IT vocabulary and reading comprehension to the algorithmization skills. Thus, the threshold value can be given as it is described in the second chapter. The differences appear clearly at a higher level of algorithmization at execution, decoding, and interpretation.

The development of the algorithmization skills has a filter function on the development of the elements of programming languages and IDE, and programming. There is a significant difference between those, whose output was below 60% compared to those who performed better. The proper usage of the elements of programming languages goes hand in hand with the development of programming.

Our results together prove those methodological recommendations which, are based on experience, suggest that forming algorithmization skills and parallel learning of IT vocabulary is a pre-requirement of successful programming. It is important that our results described in the present study do not take the development of general learning skills and motivation into consideration. We would like to continue our research with the consideration of these factors.

Acknowledgements

This examination is the part of the secondary school research programme of the first author. Authors express their special thanks to cooperating institutions: pupils, undergraduates, teachers and lecturers of Bányai Júlia Secondary Grammar School of Kecskemét; Bolyai János Secondary Grammar School of Kecskemét; GAMF Faculty of Engineering and Computer Science, John von Neumann University, Kecskemét).

This research is supported by EFOP-3.6.1-16-2016-00006 "The development and enhancement of the research potential at John von Neumann University" project. The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

References

[1] Kurland, D. M., Pea, R. D., Clement, C., Mawby, R. (1981): The Study of the Development of Programming Ability and Thinking Skills in High School Students. In: Soloway, E. & Spohrer, J. C. (eds.): *Studying the Novice Programmer*. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, USA. 83-112.

[2] Lahtinen, E., Ala-Mutka, K., Järvinen, H. (2005): A study of the difficulties of novice programmers. Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education ITiCSE '05. New York, NY, USA. 14-18.

[3] Lehrer, R., Guckengerg. T., Sancilio, L. (1988): Influences on LOGO on children's intellectual development. In: Mayer, R. E. (ed.): *Teaching and learning computer programming: Multiple research perspectives.* Lawrence Erlbaum Associates, Inc., Mahwah, New Jersey, USA. 75-110.

[4] Milne, I., Rowe, G. (2002): Difficulties in Learning and Teaching Programming – Views of Students and Tutors. *Education and Information Technologies*, 7 (1) 55–66.

[5] Pap-Szigeti, R. (2018): Szoftver- és mérőeszköz-fejlesztés programozási ismeretek és készségek értékelésére (Development of a software and tests for the evaluation of programming knowledge and skills). *Gradus*, 5 (2) 355-359. http://gradus.kefo.hu/archive/2018-2/2018_2_CSC_022_Pap-Szigeti.pdf

[6] Pásztor,A., Pap-Szigeti, R., Török, E. (2010): Effects of Using Model Robots in the Education of Programming. *Informatics in Education*, 9 (1) 133–140. https://doi.org/10.15388/infedu.2010.08

[7] Pea, R. D., Kurland, D. M. (1984): On the cognitive effects of learning computer programming. *New Ideas in Psychology*, 2 (2) 137-168.

[8] Robins, A., Rountree, J., Rountree, N. (2003): Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13 (2) 137-172.

[9] Török, E., Brindza, A., Hercz, M., Koltói, L., Pap-Szigeti, R., Rigó, R. (2016): TALENTUM: hallgatói sikerességet támogató és tanulást segítő pilot program a Kecskeméti Főiskolán (TALENTUM: a pilot program for supporting students' success in learning at Kecskemét College). In: Fodorné Tóth, K. (ed.): *Felsőoktatási kihívások. Alkalmazkodás stratégiai partnerségben.* MELLearN Felsőoktatási Hálózat az életen át tartó tanulásért Egyesület, Pécs, Hungary. 76-85.
[10] Török, E., Pap-Szigeti, R. (2018): Input survey on competence aiming to support students' efficacy. Central European Higher Education Cooperation (CEHEC) 4th Conference: Paper presented at Higher Education in Central and Eastern Europe: National, Regional and European Trajectories. Book of abstracts. CEU, Budapest, Hungary, 26-27.

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665