© 2019 WIETE

Effect of collaborative problem solving assisted by advance organisers and cognitive style on learning outcomes in computer programming

Dyah Susilowati, I Nyoman S. Degeng, Punaji Setyosari & Saida Ulfa

State University of Malang
Malang, Indonesia

ABSTRACT: The aim of this research was to examine the effects of advance organisers (AO)-assisted collaborative problem solving (CPS) on the conceptual understanding and application of computer programming for students with different cognitive styles. The research subjects were university students of information technology, 42 in an experimental group and 40 in a control group. The research findings show that AO-assisted CPS has a significant influence on the improvement of conceptual understanding and conceptual application, compared to the CPS without AO. Students' cognitive style had a significant influence on learning outcomes; students with a field-independent (FI) cognitive style have better computer programming conceptual understanding compared to students with a field-dependent (FD) cognitive style. There was no interaction effect between CPS with AO and CPS without AO with FD and FI cognitive styles, on students' computer programming conceptual understanding. There was an interaction effect between CPS with AO and CPS without AO with FD and FI cognitive styles, on students' computer programming conceptual application.

INTRODUCTION

Computer Programming is among the more challenging and daunting of courses in computer science. This is because, when learning computer programming, students need analytical, logic, mathematics and problem-solving skills, as well as knowledge of the programming language syntax [1]. Because of these skill requirements, learning computer programming increases the cognitive load [2].

The difficulty of computer programming is one of the seven challenges that students confront in the study of computer science and the instructional strategy needs to take this into account to improve learning outcomes [3]. The concept of programming is the foundation for thinking and communicating about computer programming [4]. Conceptual understanding is also a key aspect of learning [5], since learners employ conceptual understanding when solving problems. Weaknesses in understanding the basic concepts of programming cause difficulty for students in computer programming.

Understanding is obtained when students are able to relate and integrate newly acquired knowledge with what they have previously learned. In learning, it is important to emphasise the conceptual understanding, which serves as the foundation for using knowledge [6]. Learning outcomes reflect Bloom's taxonomy of cognition, specifically cognitive dimension 2 (C2) and cognitive dimension 3 (C3). The learning outcomes that were measured in this research included student capability for presenting, classifying, grouping, summarising, comparing and explaining the concept of programming. Meanwhile, the conceptual application was based on Bloom's cognitive process dimension involving implementation and execution of programs. This also includes a student's capacity to apply programming concepts. For instance, when learning recursion, learners should then be trained with problems related to the concept of recursion, such as the base case and Hanoi tower.

Instructional design is a suitable strategy by which to teach students to solve programming problems. A problem-solving method increases the cognitive load for learners [7], but allows learners to solve basic problems. However, when exercises are complex, they can cause an excessive cognitive load and lead to frustration. Collaborative problem solving can ease the cognitive load of an individual. Learning collaboratively is more effective than individual learning for problem solving [7].

A learner's problem-solving skills can be improved significantly by collaborative instructional learning [8]. Collaborative instructional learning can boost students' confidence, increase their involvement, and improve their attitude, as well as developing co-operation skills when working in a team, which will be useful for future professional work [9].

A collaborative problem-solving instructional strategy involves two or more participants. Collaborative problem solving is an important 21st Century skill and a key success factor [10]. To optimise collaborative problem solving, students' conceptual understanding must be strengthened. Scaffolding can be used to improve a learner's conceptual understanding of computer programming. Scaffolding produces a preliminary structure as a foundation, which helps build understanding [11][12]. This approach helps learners to acquire computational concepts by providing a structure for the student to build on.

One of several strategies to promote learning is called advance organiser (AO). The AO model assists instruction by providing a conceptual structure and information assimilation [13]. The AO is a pedagogical tool that supports the progressive differentiation of knowledge during learning and the integrative reconciliation of knowledge to bridge students' current knowledge and the knowledge they might need in the future [14].

The AO is very useful in a situation where the learner has not attained relevant conceptual knowledge by the time material is presented to the learner [15]. The use of AO improves the learning outcomes for computer programming [16]. In this research, AO was combined with problem solving regardless of a learner's characteristics, such as cognitive style and motivation, which also affects learning outcomes. It was found through this research that AO could improve learning outcomes [16-25].

To achieve learning outcomes, it is crucial to choose an instructional strategy, which takes into account the cognitive style of the learner. Learners with different cognitive styles can have different learning processes, which results in different learning outcomes. Cognitive style is defined as a psychological construction related to the preferred way an individual processes information [26] and it is classified into field-dependent and field-independent categories [27].

Cognitive style is closely related to the choice of instructional strategy and so, there exists a correlation between cognitive style and instructional strategy [28]. Learners with a field-independent learning style have more potential to succeed in learning computer programming, while learners with a field-dependent learning style need additional support to learn computer programming [29-32]. Cognitive style has a more significant correlation with learning during the design stage than during the coding stage [33][34].

There have been previous studies of instructional strategy in computer programming. However, there are few studies that examine the impact of instructional strategy collaborative problem solving and cognitive style on learning. This research was conducted to study the effectiveness of collaborative problem solving with advance organiser and cognitive style on conceptual understanding and conceptual application. The following hypotheses were examined:

Hypothesis 1: there is a significant difference in the conceptual understanding of computer programming between a group of students with advance organiser-assisted collaborative problem solving (CPS with AO) and a group of students with collaborative problem solving without advance organiser (CPS without AO).

Hypothesis 2: there is a significant difference in conceptual application between a group of students with CPS with AO and students with CPS without AO.

Hypothesis 3: there is a significant difference in the conceptual understanding of computer programming between a group of students with field-dependent cognitive style (FD-CS) and a group of students with field-independent cognitive style (FI-CS).

Hypothesis 4: there is a significant difference in the conceptual application of computer programming between a group of students with FD-CS and students with FI-CS.

Hypothesis 5: there is an interaction between instructional strategy (CPS with AO and CPS without AO) and cognitive styles (FD-CS and FI-CS) and the conceptual understanding of computer programming.

Hypothesis 6: there is an interaction between instructional strategy (CPS with AO and CPS without AO) and cognitive styles (FD-CS and FI-CS) and the conceptual application of computer programming.

RESEARCH METHOD

Experimental Design

The research design was quantitative with a quasi-experimental pre-test/post-test approach with non-equivalent control and experimental groups [35]. The independent variables in the research were two types of instructional strategy, viz. advance organiser-assisted collaborative problem solving and collaborative problem solving without advance organiser. The dependent variables were conceptual understanding and conceptual application. In addition, studied in this research were several moderator variables (dependent field and independent field).

Research Subjects

The research involved two groups of students: an experimental group (class) and a control group (class). The subjects of the study were students of Information Technology Engineering of STMIK Bumigora Mataram, West Nusa Tenggara. The number of students involved was 82, with 40 students in the control group and 42 students in the experimental group.

Treatment Procedure

In the research, there were two treatment models for the two classes, i.e. experimental and control. The experimental class was treated using advance organisers-assisted collaborative problem solving, while the control class received treatment in the form of collaborative problem solving without advance organisers. Student grouping in the study was based on the instructional strategies used and the students' cognitive style.

The treatment procedure is presented in Table 1. The treatment was done seven times, in addition to one pre-test and one post-test. The programming materials used in the study include:

1) branching structures;
2) loop structures;
3) arrays;
4) recursion.

Table 1: Learning treatment procedures.

Step	Collaborative problem-solving strategy (CoPS)	Advance organiser (AO)-assisted CoPS
1	Lecturers select groups (two per group)	Lecturers select groups (two per group)
2	Lecturer assigns tasks through practical worksheet (LKP)	Lecturer assigns tasks through practical worksheet (LKP)
3	Without AO	Lecturer gives course introduction using AO
4	Students work on the tasks in groups through the following stages: identify the problem, problem representation, compile the algorithm, implement the algorithm (coding/program), record all co-operation processes/discussions	Students work on the tasks in a group through the following stages: identify the problem, problem representation, compile the algorithm, implement the algorithm (coding/program), record all co-operation processes/discussions
5	Conclusion	Conclusion

Research Instrument

Three types of instrument were used to measure the research variables; namely:

1) cognitive style test;
2) concept comprehension test;
3) concept application ability test.

Students' cognitive style was measured using the group embedded figure test (GEFT) [27] that distinguishes two cognitive styles: field-independent and field-dependent. This test was performed before the subjects were treated together with the pre-test.

The measurement of conceptual understanding and conceptual application was carried out using a comprehension test on computer programming conceptual understanding and conceptual application developed by the researchers. The result of the test was then discussed with the lecturers of the computer programming course. The test of concept comprehension and test of concept application ability were administered in the form of test essays. Problems in the pre-test and post-test were different, but had equal levels of difficulty.

In order to maintain the validity and reliability of the instrument developed by the researchers, every question item in the pre-test and post-test was measured using content validity. Content validation of the concept comprehension test and concept application ability test were performed by three content learning experts who have expertise in the field of computer science and informatics. The content learning experts used an instrument, which consisted of: lattice questions; concept comprehension tests; application programming concept tests, and assessment rubrics for review and validation. The reliability test used Cronbach's alpha coefficient and was performed using SPSS software.

FINDINGS

An analysis requirements test was conducted to determine the parametric feasibility prior to hypothesis testing. The analysis requirement test for univariate or multivariate analysis consists of normality and homogeneity tests.

Conceptual understanding and application scores from the statistical test result output table indicated that its significant value (probability) for advance organiser-assisted collaborative problem solving (CPS with AO) was 0.380 ($p > 0.05$) and for collaborative problem solving without advance organiser (CPS without AO) was 0.220 ($p > 0.05$). It means that in both sets of data, the score value of conceptual understanding and conceptual application of computer programming (post-test) in the experiment and control classes had a normal distribution.

Based on the Levene test result, the significance value for conceptual understanding was 0.089, which is higher than alpha 0.05 ($p > 0.05$). This leads to the conclusion that the variance of conceptual understanding data was homogenous. The same applies to the conceptual application data, the significance value of which was 0.184, higher than alpha 0.05 ($p > 0.05$). It can be inferred that the variance of conceptual application data was homogenous.

Presented in Table 2 the mean value of conceptual understanding for the experimental class (CPS with AO) was 51.36, higher than the control class (CPS with AO) of 42.24. The mean value of conceptual application in the experimental class was 82.64, which was higher than the control class of 69.79.

Table 2: Mean value of conceptual understanding and application for the control and experimental classes based on instructional strategies.

Dependent variable	Instructional strategies	Mean	Std error	95% Confidence interval	
				Lower bound	Upper bound
Concept value post	Control (CPS without AO)	42.24	2.073	38.112	46.365
	Experiment (CPS with AO)	51.36	1.950	47.479	55.241
Application value post	Control (CPS without AO)	69.79	1.804	66.200	73.384
	Experiment (CPS with AO)	82.64	1.697	79.261	86.017

Analysis

Presented in Table 3 are the results of a MANOVA 2 x 2 analysis on the influence between factors based on instructional strategies (CoPS with AO and CoPS without AO) and cognitive style on conceptual understanding and application.

Table 3: Tests of between-subjects effects.

Source	Dependent variable	Type III sum of squares	df	Mean square	F	Sig.
Corrected model	Conceptual understanding	10449.133a	3	3483.044	22.274	0.000
	Conceptual application	9326.708b	3	3108.903	26.240	0.000
Intercept	Conceptual understanding	169197.126	1	169197.126	1082.005	0.000
	Conceptual application	448746.807	1	448746.807	3787.568	0.000
Instructional strategies	Conceptual understanding	1606.847	1	1606.847	10.276	0.002
	Conceptual application	3187.585	1	3187.585	26.904	0.000
Cognitive style	Conceptual understanding	6809.426	1	6809.426	43.546	0.000
	Conceptual application	3541.466	1	3541.466	29.891	0.000
Instructional strategies*	Conceptual understanding	1.942	1	1.942	0.012	0.912
Cognitive style	Conceptual application	523.996	1	523.996	4.423	0.039
Error	Conceptual understanding	12197.146	78	156.374		
	Conceptual application	9241.352	78	118.479		
Total	Conceptual understanding	197585.362	82			
	Conceptual application	483578.695	82			
Corrected total	Conceptual understanding	22646.279	81			
	Conceptual application	18568.061	81			

Note: S = significant; NS = not significant; $\alpha = 0.05$

DISCUSSION

In Table 3, it is shown that conceptual understanding had F = 10.276 with a significance of 0.002, which is lower than alpha 0.05. This shows that there was an effect of CPS with AO and CPS without AO on the conceptual understanding of computer programming. In addition, there was a significant difference in the mean value of the conceptual understanding of computer programming, where students who learn using CPS with AO had a higher mean at 51.36 (see Table 2) than those who learn using CPS without AO at 42.24. Therefore, CPS with AO is better than CPS without AO in improving students' conceptual understanding of computer programming. Moreover, the conceptual application of computer programming in Table 3, F = 26.903 with a significance value of 0.000 is lower than alpha 0.05. This means that there was a significant effect of CPS with AO and CPS without AO on conceptual application.

There was also a significant difference of the mean value of the conceptual application of computer programming, where students who learn using CPS with AO had a higher mean than those who learn using CPS without AO, which were 82.64 and 69.79, respectively (see Table 2). Therefore, CPS with AO is superior compared to CPS without AO in helping students in the application of computer programming. The result is in accordance with several previous research results on the use of advance organisers and their influence on learning outcomes [21][35]. Indicated by the research results, the advance organisers could strengthen the cognitive structure and facilitate learners' acquisition of new material [14]. In addition, Mayer found advance organisers effective in improving learning outcomes [15].

The F statistic value for the conceptual understanding of computer programming based on cognitive style was 43.546 with significance value of 0.000, which is lower than alpha 0.05. There was a significant difference on conceptual understanding between the students who had a field-dependent (FD) cognitive style and students who had a field-independent (FI) cognitive style. Students with field-independent (FI) cognitive style had better conceptual understanding of computer programming than those with a field-dependent (FD) cognitive style. This finding is supported by the significant difference on post-test mean values of conceptual understanding with field-dependent (FD) cognitive style, which was 36.55, and field-independent (FI) cognitive style, which was 57.35.

Referring to Table 3, the F value for the conceptual application based on field-dependent (FD) and field-independent (FI) cognitive styles was 29.890 with a significance value of 0.000, which is smaller than alpha 0.05. Therefore, there was a significant difference on the conceptual application of computer programming between students who have a field-dependent (FD) cognitive style and students who have a field-independent (FI) cognitive style. This result was supported by post-test mean values for students with a field-independent (FI) cognitive style of 83.99 that was higher than those students with a field-dependent (FD) cognitive style, which was 67.80. The results show that students who have a field-independent learning style can achieve better conceptual understanding and application than those who have a field-dependent learning style. This finding is in line with the findings of several previous studies, which found that cognitive style influences learning outcomes [36]. The finding also supports the claim that students need the appropriate cognitive style to succeed in programming [37] and that learning programming requires some prerequisites [38]. Cognitive style influences computer programming learning [32].

In Table 3, it is shown that there was no significant difference on conceptual understanding of computer programming between the students with CPS with AO and cognitive style (field dependent FD and field independent FI) and the students with CPS without AO and cognitive style. For this case, F = 0.012 with significance value of 0.912, which is higher than alpha 0.05. The mean value of conceptual understanding of the students with CPS with AO was higher than those who have field-dependent (FD) or field-independent (FI) cognitive style. The MANOVA test results indicate there was no interaction effect between CPS with AO and CPS without AO with field-dependent (FD) and field-independent (FI) cognitive styles on students' conceptual understanding of computer programming. There are many internal characteristics that affect learning, such as motivation, self-regulation learning, self-efficacy and metacognition.

In Table 3, it was shown that there was an interaction effect between CPS with AO and CPS without AO, with field-dependent (FD) and field-independent (FI) cognitive styles on students' conceptual application of computer programming; F = 4.423 with significance value of 0.039, which is lower than alpha 0.05. In other words, students with CPS with AO and CPS without AO with field-dependent and field-independent cognitive styles had a different value for the conceptual application of computer programming.

The post-test mean value of conceptual application of computer programming for students with CPS with AO and field-dependent (FD) cognitive style of 78.84 was lower than that for students with CPS with AO and field-independent (FI) cognitive style of 86.80. In addition, the post-test mean value for conceptual application of computer programming for students with CPS without AO and field-dependent cognitive style was 60.42, which was lower than that for students with CPS without AO and field-independent cognitive style of 79.17.

The MANOVA test result indicated that there was an interaction effect between CPS with AO and CPS without AO with cognitive styles (field-dependent and field-independent) on students' conceptual application of computer programming. This finding agrees with previous research results that cognitive style and instructional strategy significantly influence the knowledge of learners who have field-independent characteristics and they show better knowledge in the field of integrated science than do learners who have field-dependent characteristics [31]

CONCLUSIONS

The following conclusions may be drawn from the discussion:

1. Advance organisers-assisted collaborative problem-solving instructional strategy is better in improving conceptual understanding than without advance organisers.
2. Advance organisers-assisted collaborative problem-solving instructional strategy is better in improving conceptual application than without advance organisers.
3. Students who have a field-independent learning style have better post-test results for the conceptual understanding of computer programming than students who have a field-dependent learning style.

4. Students who have a field-independent learning style have better post-test results on the conceptual application of computer programming than students who have a dependent learning style.
5. There was no interaction effect between CPS with AO and CPS without AO with field-dependent (FD) and field-independent (FI) cognitive styles for students' computer programming conceptual understanding.
6. There was an interaction effect between CPS with AO and CPS without AO with field-dependent (FD) and field-independent (FI) cognitive style on students' computer programming conceptual application.

ACKNOWLEDGEMENTS

The authors would like to thank STMIK Bumigora Mataram for the support.

REFERENCES

1. Sarpong, K.A. and Arthur, J.K., Causes of failure of students in computer programming courses: the teacher - learner perspective. *Inter. J. of Computer Applications*, 77, **12**, 27-32 (2013).
2. Castro, F.E.V.G., Investigating novice programmers' plan composition strategies. *Proc. Eleventh Annual Inter. Conf. on Inter. Computing Educ. Research*, 249-250 (2015).
3. McGettrick, A., Boyle, R., Ibbett, R., Lloyd, J., Lovegrove, G. and Mander, K., Grand challenges in computing: education - a summary. *The Computer J.*, 48, **1**, 42-48 (2005).
4. Arends, RI., *Learning to Teach*. (9th Edn), New York: McGraw-Hill (2012).
5. Santrock, J.W., *Educational Psychology*. (5th Edn), New York: McGraw-Hill (2011).
6. Anderson, L.W. and Krathwohl, D.R. (Eds), *A Taxonomy for Learning, Teaching, and Assessing*. New York: Addison Wesley Longman,Inc. (2001).
7. Retnowati, E., Ayres, P. and Sweller, J., Can collaborative learning improve the effectiveness of worked examples in learning mathematics? *J. of Educational Psychology*, 109, **5**, 1-14 (2016).
8. Kim, M.C. and Hannafin, M.J., Scaffolding problem solving in technology-enhanced learning environments (TELEs): bridging research and theory with practice. *Computers and Educ.*, 56, **2**, 403-417 (2011).
9. Burns, M., Pierson, E. and Reddy, S., Working together: how teachers teach and students learn in collaborative learning environments. *Inter. J. of Instruct.*, 7, **1**, 17-32 (2014).
10. Griffin, P. and Care, E. (Eds), *Assessment and Teaching of 21 Century Skills. Methods and Approach*. Springer (2015).
11. Caspersen, M.E. and Bennedsen, J., Instructional design of a programming course: a learning theoretic approach. *Proc. Third Inter. Workshop on Computing Educ. Research*, 111-122 (2007).
12. Mead, J., Gray, S., Hamer, J., James, R., Sorva, J., Clair, C.St. and Thomas, L., A cognitive approach to identifying measurable milestones for programming skill acquisition. *Proc. Working Group Reports on ITiCSE on Innov. and Technol. in Computer Science Educ.*, 182-194 (2006).
13. Joyce, B., Weil, M. and Calhoun, E., *Models of Teaching.* (8th Edn), New Jersey, USA: Pearson Education (2009).
14. Ausubel, D.P., *The Acquisition and Retention of Knowledge: a Cognitive View*. Springer (2000).
15. Mayer, R.E., Some conditions of meaningful learning for computer programming: advance organizers and subject control of frame order. *J. of Educational Psych.*, 68, **2**, 143-150 (1976).
16. Chen, B., Effects of advance organizers on learning and retention from a fully Web-based class. *ProQuest Dissertations and Theses*, 155-155 (2007).
17. Chuang, H.H. and Liu, H.C., Investigating the effect of different verbal formats of advance organizers on third graders' understanding of heat transfer concept. *Inter. J. of Educ. in Mathematics, Science and Technol.*, 2, **1**, 78-84 (2014).
18. Githua, B.N. and Nyabwa, R.A., Effects of advance organiser strategy during instruction on secondary school students' mathematics achievement in Kenya's Nakuru district. *Inter. J. of Science and Mathematics Educ.*, 6, **3**, 439-457 (2008).
19. Huang, H., Cheng, P.-J. and Hu, H.-J. Establishing the structure of interactive touch advance organizer for school-age children. *Proc. 7th Inter. Conf. on Computing and Convergence Technol.*, 1264-1267 (2012).
20. Korur, F., Toker, S. and Eryılmaz, A., Effects of the integrated online advance organizer teaching materials on students' science achievement and attitude. *J. of Science Educ. and Technol.*, 25, **4**, 628-640 (2016).
21. Nazimuddin, S.K., Effect of advance organizer model (AOM) on pupil's academic achievement in geography - a study. *Inter. J. of Scientific Engng. and Research*, 3, **7**, 18-23 (2015).
22. Diptoadi, V.L., Effects of prereading strategies on efl reading by indonesian college students having different characteristics. *Paper Presented at the Regional Language Centre Seminar* (1991).
23. Wachanga, S.W., Arimba, A.M. and Mbugua, Z.K., Effects of advance organizer teaching approach on secondary school students' achievement in chemistry in Maara District, Kenya. *Inter. J. of Social Science and Interdisciplinary Research*, 2, **6**, 24-36 (2013).
24. Walia, P. and Walia, P., Effect of integrated syntax of advance organizer model and inductive thinking model on attitude towards mathematics and reaction towards integration of models. *Inter. J. of Educationand Psychological Research*, 3, **2**, 15-20 (2014).
25. Brown, E., Brailsford, T., Fisher, T., Moore, A. and Ashman, H., Reappraising cognitive styles in adaptive Web applications. *Proc. 15th Inter. Conf. on World Wide Web*, 327-335 (2006).

26. Witkin, H.A., Moore, C.A., Goodenough, D.R. and Cox, P.W., Field-dependent and field-independent cognitive styles and their educational implications. *Review of Educational Research*, 47, **1**, 1-64 (1977).
27. Shi, C., A study of the relationship between cognitive styles and learning strategies. *Higher Educ. Studies*, 1, **1**, 20-26 (2011).
28. Oh, E. and Lim, D., Cross relationships between cognitive styles and learner variables in online learning environment. *J. of Interactive Online Learning*, 4, **1**, 53-66 (2005).
29. White, G.L. and Sivitanides, M.P., A theory of the relationships between cognitive requirements of computer programming languages and programmers' cognitive characteristics. *J. of Infor. Systems Educ.*, 13, **1**, 59-68 (2002).
30. Dowlatabadi, H. and Mehraganfar, M., The causal correlation between field-dependence/independence cognitive style and vocabulary learning strategies among Iranian EFL learners. *Inter. J. of Latest Research in Science and Technol.*, 3, **4**, 97-103 (2014).
31. Agboghoroma, T.E., Interaction effects of cognitive style and instructional mode on students' knowledge of integrated science. *European J. of Research and Reflection in Educational Sciences,* 3, **1**, 47-54 (2015).
32. Mancy, R. and Reid, N. Aspects of cognitive style and programming. *Proc. 16th Workshop of the Psychology of Programming Interest Group*, 1-9 (2004).
33. Bishop-Clark, C., Cognitive style and its effect on the stages of programming. *J. of Research on Computing in Educ.*, 27, **4**, 373-386 (1995).
34. Tuckman, B.W. and Harper, B.E., *Conducting Educational Research.* (6th Edn), United Kingdom: Rowman & Littlefield Publishe, Inc. (2012).
35. Matthews, R., Soon, H. and Ah, K., Practical use of review question and content object as advanced organizer for computer programming lessons. *Procedia - Social and Behavioral Sciences,* 172, 215–222 (2015).
36. Ford, M. and Venema, S., Assessing the success of an introductory programming course. *J. of Infor. Technol. Educ.*, 9, 135-145 (2010).
37. White, G., Visual basic programming impact on cognitive style of college students: need for prerequisites. *Infor. Systems Educ. J.*, 10, **4**, 74-83 (2012).
38. White, G.L. and Sivitanides, M.P., A theory of the relationships between cognitive requirements of computer programming languages and programmers' cognitive characteristics. *J of Infor. Systems Educ.*, 13, **1**, 59-68 (2002).