Anadolu Eğitim Liderliği ve Öğretim Dergisi *www.e-ajeli.com*
[Anatolian Journal of Educational Leadership and Instruction] *e-ISSN:2148-2667*
2017 – 5 (1), 17-27

MODELING THE PSYCHOLOGICAL FACTORS AFFECTING COMPUTER PROGRAMMING SELF-EFFICACY

Kağan KIRCABURUN* İbrahim BAŞTUĞ** Muhammed BAHTİYAR***

ABSTRACT: We live in such an age that computer technologies and machines are some of the most crucial parts of human beings lives that our need and dependency to them growing exponentially day by day. This leads to increased need of more human power and involvement in programming computers and machines. Despite the increasing need however, studies indicate that number of students that prefers departments regarding computer and machine programming, software developing and engineering is declining. In this direction, it is necessary to examine the factors that are related to computer programming and to investigate the reasons of developments mentioned earlier. Therefore, this study aimed to examine the psychological factors that are thought to be related to computer programming self-efficacy. In order to do that, relationships between computer programming self-efficacy, attitude toward computer programming, psychological resilience and problem solving were investigated by using correlation test and structural equation modeling. Students from four different departments and state universities filled out a questionnaire that contains Computer Programming Self-Efficacy Scale (CPSES), Attitude Scale Toward Computer Programming (ATCP), Brief Resilience Scale (BRS), Social Problem Solving Inventory-Short Form (SPSI-SF) voluntarily and anonymously. As a result of the analyses, it was observed that computer programming self-efficacy was related positively to confidence and motivation in learning computer programming, psychological resilience, positive problem orientation and rationale problem solving, and also it was negatively related to negative problem orientation. The findings were discussed in the light of existing literature.
Keywords: Computer programming self-efficacy, Attitude toward computer programming, psychological resilience, problem solving

BİLGİSAYAR PROGRAMLAMA ÖZ-YETERLİLİĞİNİ ETKİLEYEN PSİKOLOJİK FAKTÖRLERİN İNCELENMESİ

ÖZ: Günümüzde bilgisayar teknolojileri ve makineler insan oğlunun vazgeçemeyeceği en önemli olgulardan birisi haline gelmiştir ve bu ihtiyaç ve bağımlılık gün geçtikçe daha da artış göstermektedir. Bu gelişmeler de bilgisayarları ve makineleri programlayabilecek ve bu tarz görevleri yerine getirebilecek becerilere sahip insan gücüne olan ihtiyacı arttırmaktadır. Fakat, bu artan ihtiyacın aksine yapılan çalışmalar öğrencilerin bilgisayar ve makine programlama, yazılım geliştirme ve mühendisliği gibi bölümleri daha az tercih ettiklerini ortaya koymaktadır. Bu açıdan bilgisayar programlamayla ilişkili olan faktörleri ve öğrencileri daha az bu bölümleri tercih etmeye yönelten faktörleri araştırmak ihtiyaç haline gelmiştir. Bu doğrultuda bu çalışmanın amacı bilgisayar programlama öz-yeterliliği ile bilgisayar programlamaya karşı tutum, psikolojik dayanıklılık ve problem çözme arasındaki ilişkileri araştırmaktır. Dört farklı devlet üniversitesinden öğrencilerin katılımıyla yapılan çalışmada Bilgisayar Programlama Öz-yeterlilik Ölçeği, Bilgisayar Programlamaya KArşı Tutum Ölçeği, Psikolojik Dayanıklılık Ölçeği ve Problem Çözme Ölçeğinden oluşan bir form kullanılmıştır. Çalışmada gönüllülük ve gizlilik esas alınmıştır. Değişkenler arasındaki ilişkileri araştırmak için korelasyon analizi ve yapısal eşitlik modellemesi yöntemleri kullanılmıştır. Yapılan analizler sonucunda bilgisayar programlama öz-yeterliğin bilgisayar programlaya karşı tutumun alt boyutu olan programlama öğrenmede duyulan güven ve motivasyon, psikolojik dayanıklılık, problem pozitif yaklaşım ve rasyonel problem çözme ile pozitif ilişkili, ayrıca negatif problem çözme ile de negatif ilişkili olduğu sonuçlarına ulaşılmıştır. Elde edilen bulgular literatürde bulunan geçmiş çalışmalar ışığında tartışılmıştır.
Anahtar Kelimeler: Bilgisayar programlama öz-yeterliliği, bilgisayar programlamaya karşı tutum, psikolojik dayanıklılık, problem çözme.

**Ar. Gör., Düzce Üniversitesi, kircaburunkagan@gmail.com*
*** MEB, Ibrahim_bastug@hotmail.com*
**** Ar. Gör., Düzce Üniversitesi, muhammedbahtiyar@duzce.edu.tr*

INTRODUCTION

Nowadays impacts of computers and machines on human lives increasing exponentially. In parallel, this leads to increase in popularity and importance of computer programming. Therefore, need for individuals who know computer programming is growing. However, despite the growing need for computer programmers, it was observed that number of students that preferred departments regarding programming is declining rapidly in recent years (Heersink & Moskal, 2010; Patterson, 2005). Even beyond that, students enrolled in departments include computer programming were having certain difficulties in learning computer programming and being successful (Kurland et al., 1986).

Programming was defined as commands, words and arithmetic operations that direct and designate computer hardware how to behave. Another way to define programming is that it is the whole process of writing, testing and maintaining a computer program. In a wider perspective, programming is solving a complex problem by examining the problem from top to toe (Saeli et al., 2011). Learning how to program is a tough process which requires various cognitive abilities (Kurland et al., 1986). Complexity of programming and difficulty of comprehending its logic lead students to cognitive exhaustion and declining motivation toward learning programming (Kelleher & Pausch, 2005). Over the years, many studies have been conducted by action research in beginner level programming courses, however few interdiciplinary studies have been noticed (Pears et al., 2007). Reasons given earlier makes it necessary to determine all possible factors that could affect computer programming learning process. Since many previous studies in the literature have mainly focused on the affects of cognitive features of the students on computer programming abilities, this study considered the psychological factors that were thought to be influential on learning and succeeding computer programming. Therefore, purpose of this study was examining the relationships between computer programming self-efficacy, attitude towards computer programming, psychological resilience and problem solving among university students.

Computer Programming Self-Efficacy

Self-efficacy first came to be as a person's self perception and understanding of him/herself whether he or she could be successful in a certain task (Özçelik & Kurt, 2007 as cited in Şad & Demir, 2015; Askar & Davenport, 2009). Self-efficacy, which is approached by Bandura in the context of social learning theory, is for an individual to believe that he or she has the capacity to succesfully do something (Bandura, 1997 as cited in Davidsson, Larzon & Ljunggren, 2010). Self-efficacy is believed and supposed to predict individuals' behaviors (Awaidi ve Alghazo, 2012 as cited in Şad & Demir, 2015) and it is a stepping stone between individuals' knowledge and behaviors (Askar & Davenport, 2009). As much as a person has the knowlegde to do a certain task, they may not be succesful if they do not have the necessary motivation and belief (Askar & Davenport, 2009). Bandura put forward that individuals' level of beliefs to their abilities are affecting their behaviors, motivations and consequently success levels (Henson, 2001 as cited in Şad & Demir, 2015). With increase in self-efficacy, it is stated that levels of ambition, resiliency and persistence are also rising (Şad & Demir, 2015). Anxiety and stress, which are important factors that affect academic performance and achievement, may lead students to lower self-efficacy which lead further to perceive problems much more hard and inapprehensible than they actually are (Askar & Davenport, 2009; Davidsson et al., 2010). Individuals' self-efficacy is claimed to developed parallel to their inclining abilities and experiences (Bandura, 1986 akt Askar & Davenport, 2009). Individuals interpret new information according to their past experiences (Askar & Davenport, 2009).

Affects of self-efficacy to learning programming is subjected by some researchers in various studies. As a result of their study which has investigated the evolution of self-efficacy levels through programming course, Davidsson et al. (2010) reported that even though self-efficacy was not differentiated statistically, self-regulation levels of students were increased. While Askar & Davenport (2009) indicated that male students had higher levels of self-efficacy among engineering students in

Java programming courses, Ramalingam & Wiedenbeck (1998) reported that female students' levels of self-efficacy were still low at the end of the courses, however they went up through courses. Ramalingam, Belle & Wiedenbeck (2004) also found that self-efficacy was significantly influenced by the previous programming experiences (Askar & Davenport, 2009).

Attitude Toward Computer Programming

Attitude is readiness against an object or a fact (Gökdaş, 2008). Attitude affects our behaviors and how we learn (Maio & Haddock, 2009 as cited in Özyurt & Özyurt, 2015). Since it designates how they behave in certain activities, attitude of individuals toward those activities is important (Jay, Willis & Gerontol, 1992 as cited in Charters et al., 2014).

Computer programming is being used in variety of business areas as coding or interactive programming nowadays. Most of the contemporary companies consider computer efficiency as a necessity for middle and upper class executives, and also importance of being a computer literate went up (Charters, Lee, Ko ve Loksa, 2014). Despite these improvements, research indicates that number of students who have chosen computer sciences were decreasing (Charters et al., 2014; Heersink & Moskal, 2010; Hoegh & Moskal, 2009 as cited in Başer, 2013). Some of them believed that programming was very hard to learn and that they did not possess necessary abilities such as analytical thinking and intelligence, also some others considered that programmers were weird and antisocial individuals, who were obsessed with technology and lack communication skills (Charters et al., 2014; Margolis & Fisher, 2002; Rodger et. al., 2009). These elements affect students' attitude toward programming negatively and discourage them from learning programming (Charters et al., 2014). Apart from these, complex punctuation marks and designs of the programming languages used for software development also affect attitude of the students toward computer programming negatively.

These perceptions lead students to develop negative attitude toward programming and inhibit them to see programming as a necessary proficiency and to learn programming (Charters et al., 2014). On the other hand, positive attitude increases computer use frequency and understanding of sub skills (Utting, Cooper, Kölling, Maloney & Resnick, 2010). In order to decrease the negative attitude toward programming researchers and instructors organize and use computer camps and educational games among secondary and high school students (Utting et al., 2010). Programs such as Scratch, Alice and Greenfoot were also developed to increase students' positive attitude toward learning programming (Charters et al., 2014).

Previous research has shown that differences of attitude toward computers among female and male students have gone down after courses end (Shasshani, 1997). Gökalp & Aydın (2013) also reported that students' attitudes toward use of internet and computers were moderately positive and these attitudes were affected by some variables. Charters et al. (2014) indicated that students' attitudes were gone from negative to positive among adults once they had courses by using online educational game. Başer (2013) reported more positive attitude among males compared to female students and that attitude was positively related to learning achievement.

Psychological Resilience

Resilience defined as individuals' power to motivate and sooth themselves when they face hard and difficult situations. Resilient individuals have the ability to rapidly pull themselves together when they face traumatic or trajedic events (Coşkun, Garipağaoğlu, Tosun, 2014). Kobasa (1979) laid the bases of this theory (Yöndem & Bahtiyar, 2016). Resilience is an important quality because it affects students' socio-cognitive development process and make them wiser and stronger by helping them with the challenges faced (Coşkun et al., 2014). Student-centered education approach is becoming more and more important in contemporary higher education, hence social and personal differences of students are being examined and in accordance with these differences, new instructional programmes are beginning to be applied by educators (Coşkun et al., 2014). Zimmerman & Arunkumar (1994) defined resilience as staying positive and to be able adapt to new conditions in most difficult and

intimidating circumstances (Coşkun et al., 2014). In this context, resilience is closely related to psychological resilience, it is improvable and it comprise of skills which are needed to cope with tough situations (Gizir, 2007; Öz & Yılmaz, 2009 as cited in Coşkun et al., 2014). Individuals' acceptance among their peers increases psychological resilience (Criss et al., 2002). Psychologically resillient individuals believe that they have the power to direct their life, they attend actively into life and they see change as an exciting struggle (Simoni & Paterson, 1997 as cited in Yöndem & Bahtiyar, 2016). Further, resilience is considered important through the problem solving process. Individuals' ability to overcome problems they face, warrior personality and ability to adapt to new circumstances are also indicators of psychological resilience (Coşkun et al., 2014). Recent studies indicated significant strong positive link between psychological resilience and problem solving skills (Coşkun et al., 2014). Likewise, Yöndem & Bahtiyar (2016) found that adolescents' levels of psychological resilience were strongly associated with their ability and strategy to cope with stress. Also study of Li, Eschenauer & Yang (2013) asserted that resilience was related both to self-efficacy and problem solving and played a mediating role between these two. In this study resilience was deemed important by moving from the definition made by Zimmerman & Arunkumar (1994). Computer programming demands to be ready and strong against the new and difficult troubles faced because rendering a programme and finding the errors in the thousands of lines of codes is overwhelming, exhausting and psychologically corrosive. Students who do not have the necessary resilience and patience will easily give up learning and succeeding computer programming.

Problem Solving

Problem solving is one of most needed skills for human beings to survive and it is shaped by ones' surroundings (Ahmetoğlu, Ercan ve Akşin, 2016). Individuals, beginning from childhood, solve problems they are against. They organize the information they receive from their surroundings with their beliefs and consciousness and these cognitive processes and organizations lead individuals to develop different problem solving skills (Habibi & Milani, 2014). Psychologists, computer scientists and educators argue that computer programming is a vital tool in order children to improve their problem solving skills and to enhance their thinking abilities (Kurland et al., 1986). Students who are learning computer programming also gain new strategies for problem deliberation, solution and excogitation. Because programmer faces a certain problem and he/she produces a solution to that problem. In this process, individuals both need to solve the problem and find a proper way to communicate with the machine at the same time (Papert, 1980; Saeli et al., 2011). In order to make a complex programme individuals need higher cognitive and problem solving skills as much as they require knowing how programming language works (Kurland et al., 1986). Programming also requires dividing big and complex problems into smaller less complex pieces in order to find a solution (Saeli et al., 2011). In order to solve the problems, programmers plan, code, render and debug. This process requires complex cognitive skills (Kurland et al., 1986), hence problem solving skills are deemed crucial.

Previous studies have found significant relations between programming, complex cognitive skills, mathematics and reasoning skills (Kurland et al., 1986). Kukul and Gökçearslan (2014) reported that students who took programming course for the first time showed higher levels of problem solving skills. In other study, Habibi & Milani (2014) indicated that problem solving skills and programming scores were significantly positively associated. Lastly, Ahmetoğlu et al. (2016) found positive link between problem solving approach and academic achievement among pre-service teachers. This study adds to the existing literature in that it puts forward direct and indirect associations between computer programming self-efficacy, attitude towards computer programming and the sub-factors of problem solving such as positive-negative problem orientation and rationale problem solving.

Aims of the Study

The current study aimed to investigate the relations between computer programming self-efficacy (CPSE), attitude toward computer programming, psychological resilience and problem solving. In this context, a structural equation model was constructed by using CPSE, confidence and motivations in

learning programming (CMLP) sub factor of attitude toward computer programming, psychological resilience (PR), positive (PPO) and negative (NPO) problem orientations and rationale problem solving (RPS) sub factors of problem solving inventory. Based on the previous studies regarding variables of the study, hypotheses below were designated.

Hypotheses of this study are:

1- There is a positive relationship between PR and PPO.
2- There is a negative relationship between PR and NPO.
3- There is a positive relationship between PR and RPS.
4- There is a positive relationship between PPO and CMLP.
5- There is a negative relationship between NPO and CMLP.
6- There is a positive relationship between RPS and CMLP.
7- There is a positive relationship between PR and CMLP.
8- There is a positive relationship between PPO and CPSE.
9- There is a negative relationship between NPO and CPSE.
10- There is a positive relationship between RPS and CPSE.
11- There is a positive relationship between PR and CPSE.
12- There is a positive relationship between CMLP and CPSE.

METHOD

Participants

Students who have computer programming courses compose the target population of the study. The study group consists of 461 students enrolled in 4 different state university in region of Marmara and Black Sea who have taken or are currently continuing at least one programming course. Characteristics of the participants are shown in Table 1.

Table 1
Characteristics of the Participants

		Frequency	**%**
Gender	Female	180	39
	Male	281	61
Department	Computer Engineering (Faculty of Technology)	166	36
	Computer Engineering (Faculty of Engineering)	54	11.7
	Computer Programming Vocational High School	140	30.3
	Computer Education and Instructional Technology	101	22

Instruments

Computer Programming Self-Efficacy Scale (CPSES): The scale was developed by Ramalingam and Wiedenbeck (1998) and adapted to Turkish by Altun and Mazman (2012). This scale contains 9 items on a 7–point likert scale in 2 sub factors which are basic (BP) and complex programming (CP). Reliability coefficient and confirmatory factor analysis (CFA) results are given in Table 2.

Attitude Scale Toward Computer Programming: This scale was developed by Başer (2013). It consists of 4 subscales which are confidence and motivation in learning programming, usefulness of programming, attitude toward success in programming and social perception of success in programming. In this study, only "Confidence and motivation in learning programming" (CMLP) subscale was used. CMLP consists of 17 items in 5-point likert scale such as "I think i will overcome harder programming problems", "Hard programming problems are interesting for me" and "When it comes to programming i am confident about myself". Reliability coefficient and CFA results are given in Table 2.

Social Problem Solving Inventory-Short Form (SPSI-SF): SPSI-SF consists of 5 subscales in 2 subdimensions which are problem orientation and problem solving styles. Scale was developed by D'Zurilla and Nezu (2002) and adapted to Turkish by Çekici (2009). In this study, positive problem orientation (PPO), negative problem orientation (NPO) and rationale problem solving (RPS) subfactors were used. PPO is aimed to measure the level of positive approach toward problems with items such as "when my attempts to solve a problem fail at the beginning, i believe i can reach a solution if i behave persistent and do not give up easily." and "when i face a tough problem, i believe i can solve it by myself if i make enough efforts." NPO consists of negative statements regarding problem solving orientation such as "When my attempts to solve a problem fail at the beginning, i feel dissappointed.", "When i need to take an important decission i feel anxious and unsure of myself." and "Hard problems make me unhappy." Lastly, RPS is one of the problem solving style subfactors which represents being constructive, realistic and systematic against the problems faced. Although title of the SPSI-SF involves the word "social", items of the subfactors used in this study measure individuals' general attitude against general problems they are facing in their life. Reliability coefficient and CFA results of the subfactors are presented in Table 2.

Brief Resilience Scale (BRS): BRS was developed by Smith et al. (2008) and adapted to Turkish by Doğan (2015). Scale contains 6 items in one factor structure which aimed to measure the psychological resilience (PR) levels of the individuals. Reliability coefficient and CFA results are given in Table 2.

Table 2
Cronbach's Alpha Coefficients and CFA Values of the Instruments

	Reliability	**X^2/df**	**RMSEA**	**SRMR**	**GFI**	**CFI**	**NFI**	**IFI**	**AGFI**	**NNFI**
CPSES	.93	1.55	.06	.03	.95	.99	.98	.99	.90	.99
CMLP	.92	4.77	.09	.06	.88	.91	.89	.91	.84	.89
BRS	.79	1.61	.04	.02	.99	.99	.99	.99	.98	.99
PPOS	.72	2.52	.06	.03	.99	.98	.97	.98	.97	.96
NPOS	.68	1.24	.02	.02	.99	.99	.98	.99	.98	.99
RPSS	.71	4.32	.08	.03	.99	.98	.97	.98	.94	.92

Results of the reliability and CFA for each scale were around good fit and acceptable values.

Procedure

Necessary permissions for the research have been taken from the faculty administrations. All participants of the study have taken part voluntarily and anonymously. For statistical analyses, descriptives, Pearson's correlation and structural equation modeling were utilized via Amos 23.0 and Spss 23.0.

RESULTS

In this section, results of the analyses are presented. Mean scores of the scales and inter-correlation values of the variables are shown in Table 3.

Table 3
Mean Scores, Standard Deviations and Correlation Co-efficients of The Variables

Variables	**X**	**Sd**	**1**	**2**	**3**	**4**	**5**	**6**
1. CPSE	43.28	12.30	.					
2. CMLP	62.56	12.73	.75**	.				
3. PR	19.95	5.23	.36**	.35**	.			
4. PPO	18.51	3.81	.36**	.43**	.34 **	.		
5. NPO	13.57	4.32	-.28**	-.27**.	-.41**	-.31**	.	
6. RPS	18.79	3.59	.28**	.36**	.24**	.68**	-.15**	.

**p< .01

As can be seen above, all variables were significantly correlated with each other (p<.01). Also, as we look at the mean scores of the scales, all of them were higher than average except NPO. Participants of this study mostly showed high levels of CPSE, CMLP, PR, PPO and RPS. Results of the tested model, good fit and acceptable values (Hu & Bentler, 1999) are shown in Table 4.

Table 4
Model Indices

	X^2/df	RMSEA	SRMR	GFI	CFI	NFI	IFI	AGFI	NNFI
Model Results	3.03	.07	.04	.98	.98	.98	.98	.95	.96
Good Fit	< 2	<. 05	< .05	.95 <	.95 <	.95 <	.95 <	.90 <	.95 <
Acceptable Values	< 5	<. 08	< .10	.90 <	.90 <	.90 <	.90 <	.85 <	.90 <

Table 4 suggests that all of the model indices are a good fit except RMSEA. These results signify that the model is accepted. Standardized values between variables are given in Figure 1.

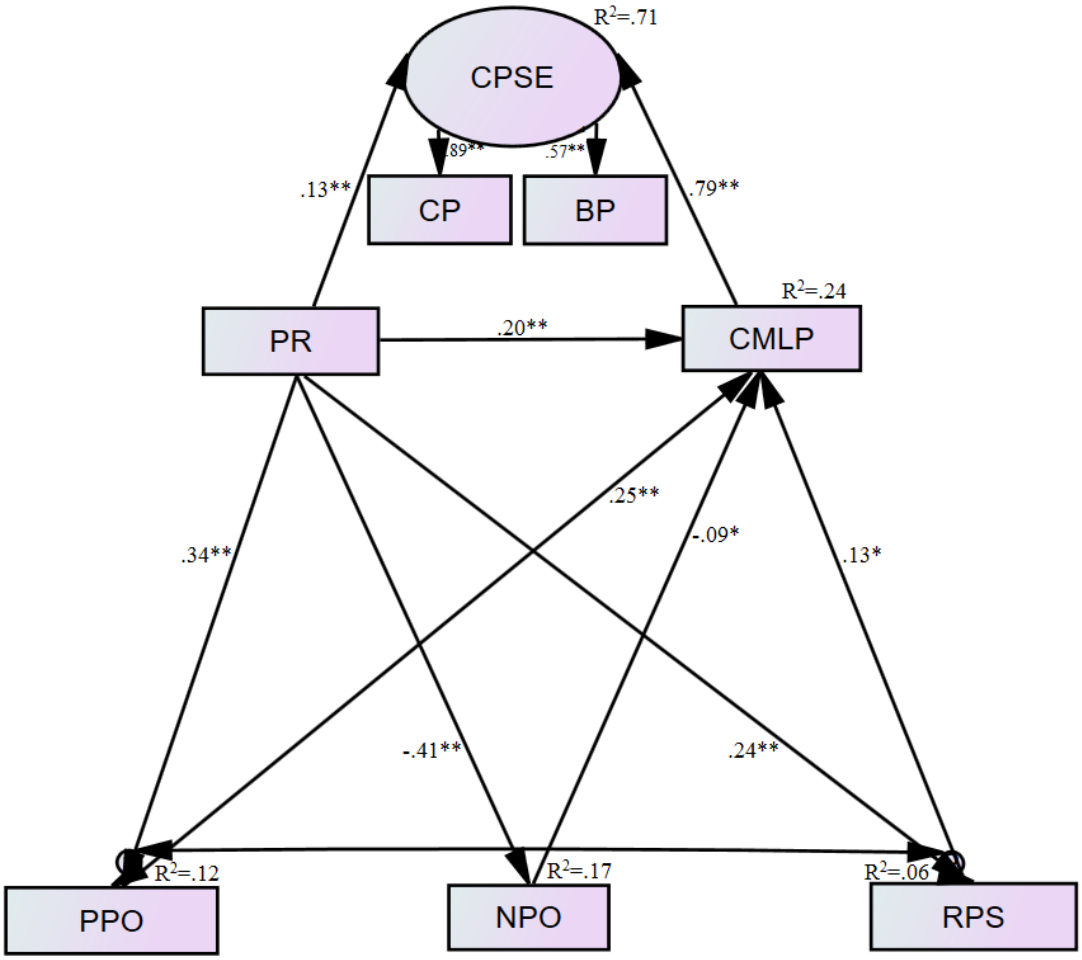

Figure 1. Standardized Values Between Variables

As demonstrated in Figure 1, there were significant relationships between PPO (t=7.84, r=.34), NPO (t= -9.56, r=-.41), RPS (t=5.21, r=.24) and PR. Also PR (t=4.28, r=.20), PPO (t=4.39, r=.25), NPO (t=-2.07, r=-.09) and RPS (t=2.36, r=.13) were directly affecting CMLP. Lastly, CPSE was directly associated with CMLP (t=11.47, r=.79) and PR (t=3.36, r=.13). All t and standardized values were significant (t>1.96, p<.05). Structural equations and R^2 values are presented in Table 5.

Table 5
Structural Equations and R^2 Values

Structural Equation	**R^2**
PPO = .34*PR	.12
NPO = -.41*PR	.17
RPS = .24*PR	.06
CMLP = .25*PPO - .09*NPO + .13*RPS + .35*PR	.24
CPSE = .79*CMLP + .41*PR + .20*PPO + .10*RPS - .07*NPO	.71

In this model, PR was able to predict %12 of PPO, %17 of NPO, %6 of RPS directly. Also it was affecting CMLP (.20 directly, .15 indirectly) and CPSE (.13 directly, .28 indirectly). When we look at the prediction of CPSE, while CMLP was associated directly with it, PR was affecting CPSE both directly and indirectly and PPO, NPO and RPS were indirect predictors of CPSE. Tested model was able to predict %24 of CMLP and %71 of CPSE.

DISCUSSION

The purpose of the current study was to investigate the relations between CPSE, CMLP, PR, PPO, NPO and RPS. Analyses revealed that CPSE was significantly related to CMLP. This finding is consistent with the study of Özyurt & Özyurt (2015) which has reported significant positive relationship between attitude toward computer programming and computer programming self-efficacy. As hypothesized, students' level of confidence and motivation in learning programming is a significant predictor of their self-efficacy on computer programming. Further, this may indicate positive relations between their attitude and their actual success and achievement in programming courses. Askar & Davenport (2009) put forward that students' computer programming efficacy beliefs was very strongly correlated with their academic achievements in programming courses. Being confident and motivated in oneself in achieving something is very important to be successful. Utting et al. (2010) indicated that positive attitude was increasing computer use frequency and understanding of sub skills. Başer (2013) further suggested that positive attitude was significantly associated with higher learning achievement.

Another finding of the study was that PR was a significant direct predictor of CPSE. Zimmerman & Arunkumar (1994) defined resilience as staying positive and to be able to adapt to new situations in most difficult and intimidating circumstances (as cited in Coşkun et al., 2014). Computer programming is a tough process that requires various cognitive abilities, patience and caution. Also it becomes very exhausting when finding small errors in thousands of lines of codes. These features may seem students discouraging and intimidating. However, as this study puts forward, students with higher psychological resilience can cope with the difficult and tiring aspects of computer programming better. Further, Yöndem & Bahtiyar (2016) reported that higher levels of PR among adolescents were related to higher ability and strategy to handle stressfull conditions. PR was also related to CPSE indirectly through PPO, NPO, RPS and CMLP. In other words, students with higher psychological resilience had higher positive problem orientation and rationale problem solving skills, lower negative problem orientation and higher levels of confidence and motivation in learning computer programming, which in turn these factors were related to higher levels of CPSE. Coşkun et al. (2014) indicated that higher PR was strongly related to higher skills of prolem solving. Psychological resilience is an important factor in dealing with problems and negative situations faced in life. This ability extends to being successful against cognitive struggles as well. Problem solving is considered to be positively associated with computer programming (Habibi & Milani (2014). Computer programming requires solving very big and complex problems by dismantling them to

smaller pieces. This is expectedly strongly related to rationale thinking and wider and positive approach to problems.

Humans are psychological beings as much as they have cognitive abilities. This study focused on the psychological factors that affect computer programming self-efficacy. Psychology is important in every aspect of individiuals' lives. No matter how much cognitive skills and abilities students have in computer programming, the authors thought that if students do not have the psychological competencies such as confidence and motivation, strong resilience, positive problem orientation and rationale thinking, they may eventually fail or underperform in computer programming. Results of the study support the notion that even a task such as learning and succeeding computer programming, which requires highly cognitive abilities, also demands higher levels of psychological qualifications as well.

This study has some limitations. Firstly, data was gathered by using self-report questionnaires, future studies should focus on the more detailed approach such as interviews. Secondly, cross-sectional nature of the study does not allow us to draw conclusion regarding causal relations. Thirdly, even though the relation between PR and RPS significant, the prediction level was rather low when compared to PR's prediction levels of PPO and NPO. This may be due to the fact that rationale thinking is rather cognitive ability which is more independent from psychological features of students such as resilience when compared to problem orientation. Lastly, results of this study is representative only of its age group which were university students, future studies may focus on the primary and secondary school students in order to generalize the findings of this study. Despite its limitations, this study has strong sides. Firstly, this study offers a model contructed of psychological factors affecting CPSE for the first time. Association of CPSE with PR, to the knowledge of the authors, has never been examined before. Lastly, study indicates more evidence on the relations between CPSE, CMLP and problem solving.

REFERENCES

Altun, A., & Mazman, S. G. (2012). [Validity and reliability analysis of Turkish form of computer programming self-efficacy scale]. *Journal of Measurement and Evaluation in Education and Psychology, 3*(2), 297-308.

Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for java programming among engineering students. *TOJET: The Turkish Online Journal of Educational Technology*, 8(1).

Başer, M. (2013). Developing attitude scale toward computer programming. *The Journal of Academic Social Science Studies, 6*(6), 199-215.

Coşkun, Y. D., Garipağaoğlu, Ç., & Tosun, Ü. (2014). Analysis of the relationship between the resiliency level and problem solving skills of university students. *Procedia-Social and Behavioral Sciences*, 114, 673-680.

Charters, P., Lee, M. J., Ko, A. J., & Loksa, D. (2014, March). Challenging stereotypes and changing attitudes: the effect of a brief programming encounter on adults' attitudes toward programming. *In Proceedings of the 45th ACM technical symposium on Computer science education (pp. 653-658). ACM*.

Criss, M. M., Pettit, G. S., Bates, J. E., Dodge, K. A., & Lapp, A. L. (2002). Family adversity, positive peer relationships, and children's externalizing behavior: A longitudinal perspective on risk and resilience. *Child development*, 73(4), 1220-1237.

Çekici, F. (2009). *The effects of skills enhancement group program which is based on problem solving therapy on college students' social problem solving skills, anger related behaviours and cognitions, trait-anxiety levels* (Doctoral Dissertation). Çukurova University, Department of Educational Sciences, Adana.

Davidsson, K., Larzon, L., & Ljunggren, K. (2010). Self-efficacy in programming among STS students. Retrieved August, 12, 2013.

Doğan, T. (2015). Adaptation of the Brief Resilience Scale into Turkish: A validity and reliability study. *The Journal of Happiness & Well-Being, 3 (1)*, 93-102.

D' Zurilla, T.J., A.M., Nezu, A, Maydeu-Olivares (2004), *Social Problem Solving: Theory and Assessment*. Chang, E.C., D' Zurilla, T.J., Sanna, L.J. (Editors), Social Problem Solving: Theory, Research and Training, American Psychological Association, Washington, DC.

Fisher, A., & Margolis, J. (2002). Unlocking the clubhouse: the Carnegie Mellon experience. *ACM SIGCSE Bulletin*, 34(2), 79-83.

Gizir, C. A. (2007). Psikolojik Sağlamlık, risk faktörleri ve koruyucu faktörler üzerine bir derleme çalışması. *Türk Psikolojik Danışma ve Rehberlik Dergisi*, 3(28), 113-128.

Gökalp, S., & Aydın, T. (2013). Teknik program öğrencilerinin bilgisayar ve internet kullanımına yönelik tutumlarının analizi. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 2(2).

Gökdaş, İ. (2008). Bilgisayara Yönelik Tutumlar. (Editör: D. Deryakulu). *Bilişim Teknolojileri Öğretiminde Sosyo-Psikolojik Değişkenler içinde* (ss.125- 150). Ankara: Maya Akademi.

Habibi, A., & Milani, M. M. R. A. (2014). Rates associated problem-solving ability with programming in computer students. *ICEMST 2014*, 854.

Hu, L. T. & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structural analysis: Conventional criteria versus new alternatives, *Structural Equation Modelling*, 6 (1), 1-55.

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, 37(2), 83-137.

Korkmaz, Ö. (2012). The impact of critical thinking and logical-mathematical intelligence on algorithmic design skills. *Journal of Educational Computing Research*, 46(2), 173-193.

Kukul, V., & Gökçearslan, Ş. (2014). Investigating The Problem Solving Skills Of Students Attended Scratch Programming Course. *8. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu Edirne.*

Kurland, D. M., Pea, R. D., Clement, C., & Mawby, R. (1986). A study of the development of programming ability and thinking skills in high school students. *Journal of Educational Computing Research*, 2(4), 429-458.

Li, M. H., Eschenauer, R., & Yang, Y. (2013). Influence of efficacy and resilience on problem solving in the United States, Taiwan, and China. *Journal of Multicultural Counseling and Development*, 41(3), 144-157.

Fisher, A., & Margolis, J. (2002). Unlocking the clubhouse: the Carnegie Mellon experience. *ACM SIGCSE Bulleti*n, 34(2), 79-83.

Özyurt, Ö., & Özyurt, H. (2015). Bilgisayar Programcılığı Öğrencilerinin Programlamaya Karşı Tutum ve Programlama Öz Yeterliklerinin Belirlenmesine Yönelik Bir Çalışma. *Eğitimde Kuram ve Uygulama*, 11(1), 51-67.

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., ... & Paterson, J. (2007). A survey of literature on the teaching of introductory programming. *ACM SIGCSE Bulletin*, 39(4), 204-223.

Ramalingam, V., & Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research*, *19*(4), 367-381.

Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2011). Teaching programming in secondary school: a pedagogical content knowledge perspective. *Informatics in Education-An International Journal*, (Vol 10_1), 73-88.

Shashaani, L. (1997). Gender differences in computer attitudes and use among college students. *Journal of Educational Computing Research*, 16(1), 37-51.

Smith, B. W., Dalen, J., Wiggins, K., Tooley, E., Christopher, P., & Jennifer Bernard, J. (2008). The brief resilience scale: Assessing the ability to bounce back. *International Journal of Behavioral Medicine*, *15*, 194–200.

Şad, S. N., & Demir, O. (2015). Computer and Internet Use Self-Efficacy Scale for Elementary School Teachers: Validity and Reliability Studies. *Elementary Education Online*, *14*(2), 489-510.

Tabanao, E. S., Rodrigo, M. M. T., & Jadud, M. C. (2011, August). Predicting at-risk novice Java programmers through the analysis of online protocols. *In Proceedings of the seventh international workshop on Computing education research (pp. 85-92)*. ACM.

Utting, I., Cooper, S., Kölling, M., Maloney, J., & Resnick, M. (2010). Alice, greenfoot, and scratch – a discussion. *ACM TOCE*, *10*(4), 17.

Wikipedia, 2016. Programming. Retrieved from https://tr.wikipedia.org/wiki/Programlama

Yöndem Z. D., & Bahtiyar M. (2016). Ergenlerde psikolojik dayaniklilik ve stresle baş etme. *The Journal of Academic Social Science Studies, 45,* 53-62.