

Development and verification of the programming resilience scale for university students

Qian Fu, Li-Wen Zhang, Jon-Chao Hong^b, Yan Dong

Beijing Normal University, ^bNational Taiwan Normal University

Learning programming challenges students who encounter difficulties such as resolving syntax and format errors. These challenges require students to invoke resilience to overcome programming problems and keep trying. In response, this study developed the programming resilience scale for university students (PRSUS). The snowball-sampling method was used to collect the data of science and engineering undergraduates, and the participants were divided into two groups ($n_1 = 316$, $n_2 = 358$) for an exploratory and a confirmatory factor analysis, respectively. For the PRSUS, 3 items were retained for each of 4 dimensions, namely: persistence, perceived value, difficulty cognition, and incremental belief. The reliability and validity indexes indicate the scale is reliable for measuring programming resilience. Moreover, the results showed that most undergraduates had a high level of programming resilience, and they could judge their level correctly in comparison to their peers. The study also found that, although no significant difference was found among the grade or age groups, the programming resilience of the male students was significantly better than that of the female students. The PRSUS is expected to help researchers and teachers to identify students who have difficulties learning programming and to provide early interventions to students.

Implications for practice or policy:

- Students can use the PRSUS to test if they have a high level of programming resilience and improve their programming resilience in a targeted way.
- Educators can use the PRSUS to identify students who find it difficult to finish programming learning or to judge whether their teaching can make students more willing to overcome programming difficulties.

Keywords: programming resilience, programming resilience scale, programming learning, academic dilemma, persistence

Introduction

Given the rapid development of intelligent technology and its growing presence in everyday lives, educational systems around the world take significant interest in programming education (Saito & Watanobe, 2020). Researchers believe that the demand for people with programming skills will increase tremendously in the future (Çetinkaya & Baykan, 2020). Many countries have incorporated programming education into their curricula (Balanskat & Engelhardt, 2014; Lindberg et al., 2019). Introductory programming courses in universities have high rates of failure and dropout: approximately one-third to one-half of students (Abdunabi et al., 2019; Margulieux et al., 2020). During the learning of programming, students usually encounter problems misunderstanding basic concepts (Tsai, 2019) or errors with symbols (Jegede et al., 2019), syntax, format, and spelling (Yarygina, 2020). In addition, in the subsequent learning process, students encounter greater difficulties and obstacles in program design (e.g., Waite et al., 2020). Thus, it is important to help students persevere in the face of programming difficulties.

Students with high academic resilience cope better with academic dilemmas and alleviate the negative impact of academic pressure. Additionally, resilience and academic achievement are positively correlated (Chisholm-Burns et al., 2019). Martin (2002) considered that, if students are not sufficiently resilient, they are more likely to fail to overcome setbacks and pressures in their studies. Thus, many researchers have developed academic resilience scales to assess whether students can cope with academic dilemmas (e.g., Cassidy, 2016; Chisholm-Burns et al., 2019; Martin & Marsh, 2006; Trigueros et al., 2020). However, these scales focus only on students' adaptations to general difficulties in learning without referring to specific subjects (Martin & Marsh, 2008). Researchers found that students showed inconsistent performance with positive and negative adaptations in different areas, accordingly performance in different subject areas varied greatly (Johnston-Wilder et al., 2013; Martin & Marsh, 2006).

As programming education gains importance, educators become more concerned about how to help students persevere when learning (Bennedsen & Caspersen, 2019). To describe science and engineering students' psychological characteristics facing across programming challenges and to understand their resilience in the context of programming learning clearly, we developed a valid and reliable programming resilience scale for university students (PRSUS). The scale can provide learners with a way to understand their programming resilience level. Moreover, the scale can help teachers judge whether their teaching is conducive to helping students persist in programming learning and attain good learning outcomes.

Literature review

Resilience in the context of programming

Research on resilience dates back to the mental hygiene movement of the 1970s. This movement showed that some children who experienced disasters or long-term difficulties did not necessarily fail in their future development as expected, but overcame the negative influence of the outside world and achieved good development (Anthony et al., 1974; Chapman et al., 2020). In 2019, researchers in South Korea began to pay attention to students' levels of resilience in programming courses. They used the concept of ordinary psychological resilience, as a personality trait (Kim et al., 2019). Researchers have not reached a consensus on the definition of resilience (Sisto et al., 2019). This research defines resilience as a personality trait in which an individual can adapt to challenges, maintain mental health, overcome difficulties, and achieve good development even in the face of adversity and pressure (Chmitorz et al., 2018; Ying et al., 2010).

Ungar emphasised in several papers that resilience must be studied in the culture and environment in which the adversity and stress arise (Ungar, 2006, 2011, 2013). Luthar (2006) also proposed that resilience should be studied in specific fields. Thus, in this paper, programming resilience is differentiated from resilience in general. This research then defines programming resilience as a personality trait where students can maintain a positive attitude and continue to engage in learning after they encounter setbacks in the process of programming learning. Accordingly, this study explored the internal structure of programming resilience, and then developed the programming resilience scale.

Measurement of programming resilience

Programming resilience is an extension of psychological resilience, and requires a related measurement structure. Garmezy et al. (1984) proposed three models for the mechanism of psychological resilience: compensatory model, the challenge model, and the protective factor model. All three models suggest that whether children can overcome adversity and achieve good development is the result of the joint action of protective and risk factors (Crandall et al., 2019). Different people have different risk factors, however protective factors are similar among those who are in trouble but develop well, and protective factors are usually easier to intervene (Martin & Marsh, 2008). Perhaps for this reason, the existing scale uses various protective factors to measure resilience.

Although many scales have been developed to measure resilience, no consensus has been reached on its underlying structure (Luthar, 2006). Hoge et al. (2007) suggested resilience scales are inconsistent, and researchers still have different views about which scale best reflects the structure of resilience. When Cassidy developed an academic resilience scale, he also pointed out the lack of an appropriate construction paradigm for measuring academic resilience, especially when taking university students as the research objects (Cassidy, 2016). Although this inherent difficulty obviously persists when developing a measurement of programming resilience, past research has yielded many items worthy of reference. From them, we extract four important dimensions of persistence, perceived value, difficulty cognition, and incremental belief, to develop the PRSUS.

Persistence

Persistence is an important internal protective factor that appears in many widely used academic resilience scales. In 1993, the Resilience Scale (RS-25) developed by Wagnild and Young (1993), and has been utilised the most among the known scales in empirical research. Wagnild and Young (1993) believed resilience consisted of five interrelated components: equanimity, perseverance, self-reliance, meaningfulness, and existential solitude. This supported the position of perseverance in the measurement

of resilience. In research on resilience in the field of education, Martin used various theories of motivation to develop the Student Motivation Scale. This scale has been revised many times, and in each revision, persistence has always been an important motivation booster (Martin, 2001, 2002, 2003). In Martin's opinion, persistence can explain students' motivation for facing academic difficulties and pressures, as well as for recovering after experiencing setbacks and continuing to struggle from the perspectives of expectancy value theory and motivation orientation.

In 2006, Martin and Marsh proposed a 5-C model of academic resilience based on these studies and developed an Academic Resilience Scale (Martin & Marsh, 2006). In the Academic Resilience Scale, commitment (persistence) plays an important role. Based on Martin and Marsh's (2006) scale, other researchers have developed more complete academic resilience scales for university students, in which persistence is a necessary dimension (Cassidy, 2016; Chisholm-Burns et al., 2019; Trigueros et al., 2020). In this study, persistence in the context of programming learning refers to the psychological trait in which, even if the programming problem students encounter is difficult or they do not receive sufficient support, they will persevere. In learning psychology, persistence is an internal protective factor.

Perceived value

As an external motivator, an individual's view of the value of an activity will largely influence his or her attitude towards the activity (Zhang et al., 2020). According to self-determination theory, if a student is aware that what they learn is useful for their own development and future career, then that awareness becomes a strong external motivation for maintaining their learning behaviour, although people's internal motivations (such as interest) do not necessarily change due to the internalisation of value (Ryan & Deci, 2000). External motivation generated by the internalisation of value is a self-determined form of motivation. Students with self-determined forms of motivation take the initiative to study, rather than choosing to study because of external forces (Deci et al., 1991). A study of third year university students suggested that those with self-determined forms of motivation are more likely to overcome difficulties in difficult courses, and persevere (Vallerand & Blssonnette, 1992). The Mathematical Resilience Scale developed by Kookan et al. (2016) also included the value dimension. Kookan et al. believed that, when students perceived the value of the content they learned, the students would think that learning behaviours would help them meet their needs. This would generate self-regulated behaviours to cope with academic dilemmas.

In this study, perceived value refers to the students' cognition of the usefulness and importance of programming knowledge and skills. If a student values programming knowledge and skills highly, and believes that learning to program is very useful, then these values and beliefs will serve as an internal protective factor for programming learning.

Difficulty cognition

Difficulty cognition reflects a student's attribution of the difficulty they have encountered. Autin and Croizet (2012) conducted three studies in which they experimented with short, psychological interventions that helped students attribute their learning difficulties to the difficulty of the material rather than to their own low intelligence. The experimental results show that reframing the attribution of difficulty can improve students' working memory efficiency within 10 minutes. It can also help them achieve higher cognitive levels and perform better on anagram tasks and reading comprehension. Moreover, it can help stop students from thinking that they are incompetent, which is beneficial in overcoming their learning dilemmas and progressing to more-challenging learning material.

In addition, identity-based motivation theory proposed by Oyserman et al. (2018) holds that an individual's perception of their identity in a certain environment will affect their sense of control, thus leading to different academic outcomes. They also suggested that teachers could help improve students' educational outcomes by interpreting academic difficulties as the importance of learning tasks and letting students know that anyone can encounter difficulties, so students do not see themselves as incompetent and remain willing to continue their efforts.

A dimension similar to difficulty cognition also appears in the Mathematical Resilience Scale developed by Kookan et al. (2016), where *struggle* was used to summarise the difficulty dimension. Kookan et al. considered that many students have difficulty learning math and sometimes must work very hard to learn it. They proposed that if these students could believe that they were facing difficulties because the content

was challenging instead of considering the difficulties a sign of their incompetence, they would have greater tolerance for setbacks. In this study, difficulty cognition refers to students' views on difficulties encountered when learning programming. If students can view programming dilemmas as a normal part of learning programming and can believe that it is normal for anyone to encounter difficulties and setbacks in the learning process, then difficulty cognition can act as an internal protective factor.

Incremental belief

According to implicit theories of intelligence, students can be divided into two categories based on their perception of their intelligence: (1) students with an entity theory of intelligence, and (2) students with an incremental theory of intelligence (Blackwell et al., 2007). The former is more likely to withdraw their efforts in the face of hardship, while the latter tend to choose more-challenging tasks and try to overcome difficulties to improve their abilities (Zhen et al., 2020). The entity theory of intelligence does harm to students' resilience, while the incremental theory promotes it (Yeager & Dweck, 2012). A study of African American university students found that convincing students that anyone's intelligence is malleable, not fixed, can help them remain committed to their studies and achieve better academic outcomes (Aronson et al., 2002). The Mathematical Resilience Scale developed by Kookken et al. (2016) also included a dimension called *growth* based on implicit theories of intelligence. Kookken et al. explained this dimension as people believing that their mathematical knowledge could be improved. However, after being translated into Chinese for this research, the meanings of the items on the scale adopted by Kookken et al., were too similar to each other, and could not be used directly in this scale.

In this study, an incremental belief refers to students' perceptions of the fact that people's ability to learn programming and their intelligence can be improved. If students realise that the level of intelligence and the ability of any person are variable, that programming is not merely for talented people, and that anyone can learn it if they try hard enough, do it the right way and get enough help, then the incremental belief acts as an internal protective factor. To summarise, four important internal protective factors of programming resilience were selected in this study to construct its scale: persistence, perceived value, difficulty cognition, and incremental belief. All four factors are related to programming learning and are based on related theories.

Purpose of the study

After considering the challenge posed to students in programming learning, programming resilience was defined. Based on the definition, the purpose of this study was to take the four aspects above as dimensions to develop and verify a programming resilience measurement tool for Chinese university students of science and engineering and to use this tool to measure the sample students' levels of programming resilience. To achieve this goal, this study was divided into three parts: (1) exploration of the factor structure of the PRSUS and verification of its reliability and validity; (2) analysis of the current situation of the programming resilience of Chinese university students of science and engineering; and (3) analysis of the differences in programming resilience between male and female students, among students of different ages, and among students of different grades.

Method

Procedure

In this study, undergraduates majoring in science or engineering completed an anonymous online questionnaire. The snowball-sampling strategy was utilised to recruit the respondents. The survey was first disseminated to students at Beijing Normal University, and they were encouraged to share the questionnaire link with other undergraduates across China. In addition, this questionnaire was disseminated by some social media platforms, such as WeChat groups, which allowed professors at other universities to forward the questionnaire link to their students. Participation in this study was voluntary. The data-collection period lasted for approximately 1 month. Of the 1,047 questionnaires were collected, 674 were valid, giving a response rate was 64.37%. All procedures followed were in accordance with the ethical standards of the responsible committee on human experimentation (institutional and national) and with the Helsinki Declaration of 1975, as revised in 2000. This study was approved by the Institutional Review Board of the author's university and informed consent was obtained from all participants for being included in the study.

Participants

The participants in this study were from 21 provinces (or municipalities directly under the central government or autonomous regions) in China. Ages ranged from 16 to 24 years old ($M = 20.28$, $SD = 1.258$). All the participants volunteered to complete the questionnaire. To facilitate the subsequent data analysis, the sample was randomly divided into two sub-samples. The grade and major distributions of the students in the two sub-samples were roughly similar. The first sub-sample consisted of 316 students, and was used for the exploratory factor analysis. The second sub-sample consisted of 358 samples, and was used for the confirmatory factor analysis. Other data analysis was performed using the total sample of 674 students. The demographic information for each sample is shown in Table 1. The full sample includes college students from different grade levels of students majoring in science and engineering. The sample comprised of 55% male and 45% female students. The distribution in the subsamples of gender, grade and major are similar to that of the total population. Therefore, the sample can be considered representative of science and engineering undergraduates in China.

Table 1
Demographic data analysis

Items	Subscale	First Sample (316 Students)		Second Sample (358 Students)		Total Sample (674 Students)	
		<i>N</i>	Proportion	<i>N</i>	Proportion	<i>N</i>	Proportion
Gender	Male	181	57.28%	194	54.19%	375	55.64%
	Female	135	42.72%	164	45.81%	299	44.36%
Grade	First year	133	42.09%	129	36.03%	262	38.87%
	Second year	108	34.18%	134	37.43%	242	35.91%
	Third year	45	14.24%	59	16.48%	104	15.43%
	Fourth year	30	9.49%	36	10.06%	66	9.79%
Major	Science	126	39.87%	132	36.87%	258	38.28%
	Engineering	190	60.13%	226	63.13%	416	61.72%

Instrument

In this study, programming resilience is divided into four dimensions. According to the definition of persistence, items with consistent meanings and non-repetition were selected from previous psychological (Lock et al., 2019) and academic resilience scales (Cassidy, 2016). The items in perceived value and difficulty cognition came from the respective dimensions of value and struggle on the Mathematical Resilience Scale developed by Kookken et al. (2016). The items in incremental belief were developed by the authors of this paper based on the definition of this dimension and several aspects of programming learning, such as mastering the basic concepts of programming and understanding the logic of a program. Each dimension had 5 items, resulting in a pool of 20 items. Table 2 shows the definitions, sources, and examples of each PRSUS construct.

Since the selected items are all from an English scale, their expressions do not necessarily conform to those of Chinese students. In addition, the states and traits expressed by these items are unrelated to programming learning, so it is necessary to adapt each item to the context of the Chinese cultural background and the context of programming learning. In the process of the adaptation, two of the authors translated the original items into Chinese, then compared the translated versions with each other, discussed the differences in the expressions and then unified these items. Then, the authors invited two international students who know nothing about resilience scales to work on the reverse translation. After the reverse translation was complete, two of the authors and the two students compared the two reverse-translated versions. These two versions aligned with one another in sense, and, in addition to the context of programming learning, the ideographic items were consistent with the original items.

Table 2
The PRSUS constructs, sources and example items

Constructs	Definitions	Sources	Example Items
Persistence (PE)	A psychological trait in which, even if the programming problem a student encounters is difficult or he or she does not get enough support, he or she won't give up but perseveres.	Cassidy, 2016 Lock et al., 2019	I am not going to lower my standards or change my goals simply because I encounter some difficulties in programming.
Perceived value (PV)	Students' cognition of the usefulness and importance of programming knowledge and skills.	Kooken et al., 2016	Having a solid knowledge of programming helps me understand more-complex issues in other disciplines.
Difficulty cognition (DC)	Students' views on difficulties encountered in programming learning.	Kooken et al., 2016	Good programmers experience difficulties when solving programming problems.
Incremental belief (IB)	Students' perception of the fact that people's programming-learning abilities and intelligence can be improved.	Self-developed	Anyone can master the basic concepts of programming through hard work.

The questionnaire posted online was divided into three parts. The first part asked about the basic personal information of the students, such as their gender, age, and grade. The grade was retested at the end of the questionnaire and served as one of the bases for the validity of the questionnaire. The second part was the items pool of the PRSUS, measured on a 5-point Likert scale (1 = *strongly disagree* and 5 = *strongly agree*). The programming resilience level is expressed by the weighted sum of all items, and the weight is determined by the proportions of variance explained of each dimension in the final scale. When the score is higher, so is the programming resilience level. The third part defined programming resilience. Students were asked to rate their own programming resilience on a 5-point Likert scale.

Data analysis

The statistical software packages SPSS 22.0 and Amos 22.0 were used for the data analysis. First, the content validity was verified by expert evaluation. Next, SPSS was used to conduct the exploratory factor analysis on the first sample and to verify the reliability of the scale. Then, Amos was used to conduct the confirmatory factor analysis on the second sample and to verify the structural validity, convergent validity, and discriminant validity of the scale. In addition, SPSS was used to derive descriptive statistics from the programming resilience level of the total sample. The measurement results of the scale were compared with the students' self-evaluation. Finally, the variance analysis was conducted by SPSS.

Results

Content validation

The content validity of the scale was tested by means of expert evaluation. According to the determined structure of programming resilience and the definitions of each dimension, two experts in the field of psychological resilience, two experts in the field of programming education, and one expert in scale

development were invited to score all the items with regard to whether the items conformed to the definition of each corresponding dimension. The rating scale adopted the form of a 4-point Likert scale (1 = *completely inconsistent* and 4 = *completely consistent*). Statistical analysis was conducted on the scoring results.

The results showed that the consistency coefficient of the experts was 1, indicating that the evaluation of the five experts was very consistent and the scale had good credibility. The average score of all the items ranged from 3.80 to 4.00, and the standard deviation ranged from .000 to .447, indicating that the programming resilience scale had good content validity.

Exploratory factor analysis

An exploratory factor analysis was performed on the first sample. Firstly, a descriptive statistical analysis was conducted on the sample data to ensure the appropriateness of these 20 items as measurement items. The standard deviation range of each item on the scale was .596 to 1.003, indicating little data fluctuation. The skewness ranged from -1.651 to -.369, within the recommended guideline of |3.0|, and the kurtosis ranged from -.156 to 4.405, within the recommended guideline of |8.0|. Therefore, the data was regarded as in normal distribution (Kline, 2011). Then, the four dimensions were extracted by principal component factor analysis, and a varimax with a Kaiser normalisation was conducted. To guarantee a pure factor measure, strong factor loadings were retained. Only those items with factor loadings larger than .60 were retained. Moreover, items with cross-loadings (3 items) were eliminated. A 4-factor structure was accepted, along with 16 items.

The Kaiser-Meyer-Olkin measure of sampling adequacy and Bartlett's test of sphericity were also conducted, together with the principal component factor analysis. The KMO coefficient was .902, which was greater than .9, and therefore interpreted as excellent (Field, 2009). In addition, the χ^2 from Bartlett's test was 2346.685 ($p < .001$). These test results indicated these items were suitable for the principal component factor analysis (Hung, 2016). The total variance explained of the remaining 16 items was 66.264%. As a social science study, a variance of more than 40% is considered sufficient (Netemeyer et al., 2003). George (2003) indicated that a Cronbach's coefficient alpha higher than .7 is considered good, and considered excellent when close to 1. The 16 items' overall Cronbach's alpha was .902, and the Cronbach's alpha of each dimension was greater than .7, showing that the scale had good internal consistency. These results indicated that the next stage of the analysis could be performed. Table 3 shows the results of the exploratory factor analysis.

Table 3
Exploratory factor analysis for the PRSUS

	IB	DC	PE	PV	Mean	SD
IB3	.856				3.80	.957
IB2	.784				4.12	.892
IB5	.782				3.73	.885
IB1	.690				3.72	.003
IB4	.679				3.76	.911
DC2		.734			4.35	.690
DC4		.731			3.91	.805
DC5		.711			4.21	.752
DC1		.670			4.60	.596
DC3		.626			3.83	.789
PE2			.771		3.61	.890
PE3			.738		3.67	.873
PE4			.724		3.64	.834
PV1				.816	3.85	.945
PV2				.726	3.83	.927
PV3				.688	3.80	.880
Variance explained	21.327%	17.839%	14.058%	13.040%		
Cronbach's alpha	.866	.806	.811	.764		

Confirmatory factor analysis

Based on the result of the exploratory factor analysis, the data from the second sample were used for the confirmatory factor analysis, and the structural validity, convergent validity, and discriminant validity of the scale were verified.

Structural validity

A confirmatory factor analysis yields a series of indexes that estimate the extent to which the sample data can fit the a priori assumptions in different ways. In the confirmatory factor analysis of this data, to achieve a high standard for the indexes and achieve as much balance as possible in the number of items under each dimension, 2 items under the dimension of difficulty cognition and 2 items under the dimension of incremental belief were removed. In the final scale, 12 items were retained. The model, composed of 12 items (Figure 1). Table 4 shows the fitting indexes corresponding to the model. The values of these indexes indicated that the scale had good structural validity.

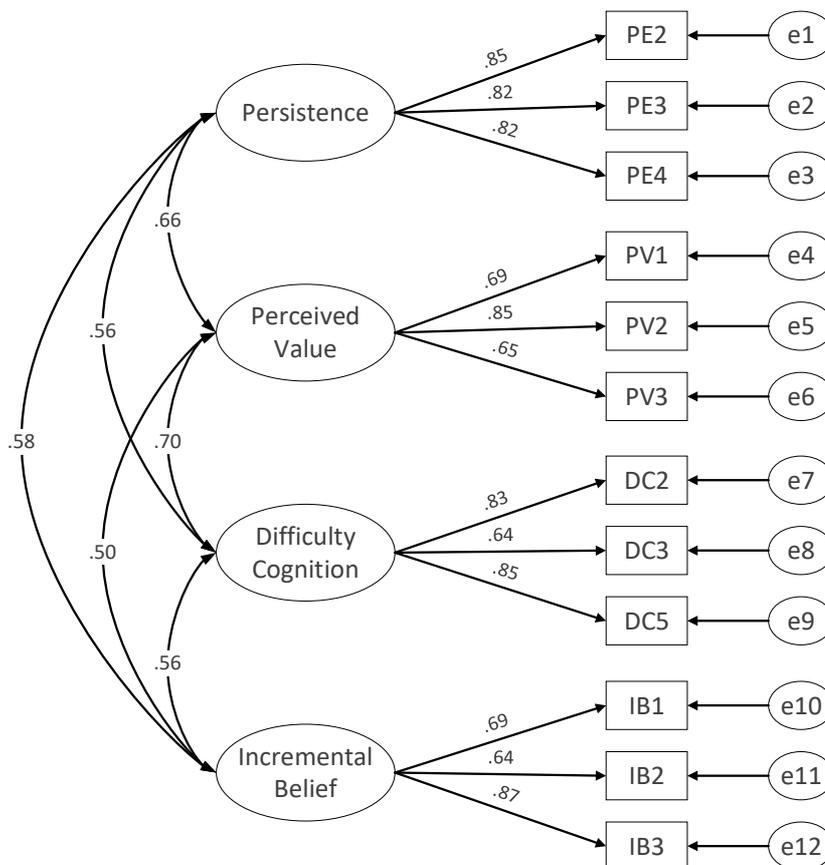


Figure 1. Confirmatory factor analysis for the PRSUS

Table 4
Fitting indexes of the scale

	CMIN/DF	GFI	AGFI	NFI	RFI	IFI	TLI	CFI	RMSEA	SRMR
Value	1.640	.964	.941	.961	.946	.984	.978	.984	.042	.024
Threshold	< 3	> .9	> .9	> .9	> .9	> .9	> .9	> .9	< .05	< .08

Convergent validity

Table 5 shows that the standardised factor loadings of all of the items were greater than 0.6, which indicates that the latent variables persistence, perceived value, difficulty cognition, and incremental belief were highly representative for these topics. Additionally, the average variances extracted (AVE) of all of the dimensions were greater than 0.5, and the composite reliability (CR) was greater than 0.7, which indicates that the scale had a good convergent validity.

Table 5
Composite reliability and convergent validity analysis

	Path		Standardised factor loading	AVE	CR
PE4	<---	PE	.817	.686	.869
PE3	<---	PE	.822		
PE2	<---	PE	.851		
DC2	<---	DC	.827	.605	.819
DC3	<---	DC	.637		
DC5	<---	DC	.852		
PV3	<---	PV	.650	.541	.777
PV2	<---	PV	.852		
PV1	<---	PV	.689		
IB3	<---	IB	.867	.546	.780
IB2	<---	IB	.644		
IB1	<---	IB	.686		

Discriminant validity

In Table 6, the values on the diagonal are the square root of the AVEs of each dimension, and the off-diagonal values are the correlation coefficient between the two dimensions. The square roots of the AVEs were greater than the absolute values of the correlation coefficients in all dimensions. This result indicated that there were both correlations and differentiation among the latent variables. Thus, the scale had an ideal discriminant validity.

Table 6
Construct discriminant validity analysis

	PE	DC	PV	IB
PE	.787			
DC	.565	.744		
PV	.663	.698	.729	
IB	.575	.556	.499	.775

In summary, the results of the data analysis above show that the scale had very good reliability and validity, so the PRSUS could be considered a feasible tool for evaluating the programming resilience of undergraduates of science and engineering. The final version of the scale is contained in Appendix A.

Programming resilience analysis

Each dimension of the scale contained the same number of items (3). By dividing the total score by 12, the resulting value, ranging from 1 to 5, was used to represent the level of the programming resilience of the students. When the score is higher, so is the level of programming resilience. The level of programming resilience of all of the 674 university students varied from 1 to 5 ($M = 3.87$, $SD = .566$). From the skewness (-.617) and kurtosis (1.875) indexes of the data, it was regarded that the data conformed to the normal distribution (Kline, 2011). The sample was divided into high and low groups based on 27% of the score, that is, less than or equal to 1.35 points was a low score, and more than or equal to 3.65 points was a high score. Approximately 70.5% of the students (475) were in the high-score group, which indicated that most of the students in the sample had an ideal level of programming resilience. However, one of the students had an extremely unsatisfactory programming resilience (5 standard deviations below the mean) and was in the low-score group.

In addition, the definition of programming resilience was given in the questionnaire so students could evaluate their own programming resilience. A Pearson correlation analysis was conducted between the self-evaluation results and the score measured by the PRSUS, and the correlation coefficient was .530 ($p < .001$). The students' scores of programming resilience under the different groups of the self-evaluation showed that, when the level of students' self-evaluation was higher, their score on the scale was also higher. Moreover, the mean value (3.81) of the students who evaluated themselves as being at the medium level was close to the mean value (3.87) of all 674 samples measured by the scale, indicating that most of the students had a correct understanding of their programming resilience.

Variance analysis

Table 7 shows the results of the independent sample *t*-test of the programming resilience of the male and female students. The data indicated that there was a significant difference in the programming resilience between the males and the females, and the programming resilience of the male students was significantly higher than that of the female students. In the independent sample *t*-test, the effect size is manifested as a mean difference such as the Cohen's *d*, which indicates the strength of the correlation between the gender and the programming resilience. According to the value of the Cohen's *d* (0.494), the difference between the male and female students had a medium effect (Lakens, 2013).

Table 7
The level of programming resilience of the gender subsample

Gender	<i>N</i>	<i>M</i>	<i>SD</i>	<i>t</i> -value	<i>p</i> -value	Cohen's <i>d</i>
Male	375	3.99	.567	6.442	0.000***	0.4937
Female	299	3.72	.526			

Note. *** $p < .001$

Moreover, according to the results of the ANOVA, no significant difference was found in the level of programming resilience between the age ($F = 1.978$, $p = .139$, $\omega^2 = .003$) or the grade groups ($F = .354$, $p = .787$, $\omega^2 = -.003$). The result of the ANOVA for age showed a small effect size, that is, age accounted for .3% of the variability in the programming resilience scores. Regarding grade, there was almost no effect size (Lakens, 2013).

Discussion and conclusion

In recent years, programming learning has attracted increasing attention. It has become an important skill for students in the twenty-first century (Durak, 2018), and it is considered helpful for developing other twenty-first century skills (Elçiçek & Karal, 2020). However, coding is difficult for many students. Ways to improve students' programming-learning participation and help them persist in the face of difficulties when programming learning have gradually become a hot topic for researchers (Pattanaphanchai, 2019).

In this research, programming resilience was defined as an adaptive personality trait, following one of the mainstream definitions of resilience (Sisto et al., 2019). In addition, the PRSUS was developed to evaluate whether university students could persist in the face of academic dilemmas when they learn programming and trying their best to overcome difficulties. In constructing the scale, we started from the mechanism of the action of resilience and used four internal protective factors. Among them, the Mathematical Resilience Scale of Kookken et al. (2016) provided two of the dimensions: perceived value (value) and difficulty cognition (struggle). The items in the dimension incremental belief were developed by the authors of this paper. In addition, the dimension persistence was included in the structure. Items in this dimension were adapted from previous academic resilience scales (Cassidy, 2016; Lock et al., 2019). Overall, the results show that the PRSUS has good reliability and validity. In particular, in terms of model fitting and the total variance explained, the index value of this scale was better than that of many academic resilience scales and those for specific disciplines (e.g., Cassidy, 2016; Kookken et al., 2016; Trigueros et al., 2020). Educators will be able to use this scale to identify those students who find it difficult to finish programming learning or to judge whether their teaching can make students more willing to overcome programming difficulties.

The PRSUS measurement showed that most of the participating Chinese undergraduates of science and engineering had a moderately high level of programming resilience, and generally had a correct understanding of their own level when compared with their peers. A similar conclusion can be seen in the study of first and second year students majoring in pharmacy, where Chisholm-Burns et al. (2019) found that the level of academic resilience was above average. However, a number of students remained who had a low level of programming resilience, who were easily defeated by the difficulties in programming learning, and who tried to avoid programming courses. This type of student needs additional scaffolding tools and encouragement from his or her teachers to overcome difficulties, develop self-confidence, and continue learning.

In addition, the programming resilience of male students was significantly higher than that of female students, and this gender difference was not related to the age or grade. This was not consistent with similar findings about academic resilience. When studying mathematical resilience in 7th and 8th-grade students, Ricketts et al. (2017) found that girls had greater resilience than boys. The results of Cassidy's (2016) research about academic resilience showed that, for British undergraduates, gender resulted in small but insignificant differences. Research has shown that the dominant western perception of computer science, especially programming, is that it is a masculine endeavour (Keller & John, 2020). Female students tend to lack the curiosity, motivation and integration capabilities for programming, and they lack the confidence in their ability to code properly (Malik & Coldwell-Neilson, 2018). This might be one reason why boys have higher levels of programming resilience than girls. Therefore, girls deserve more attention from teachers in learning, regardless of the girls' age or grade, which means helping female students improve their programming resilience will help improve their programming-learning performance and outcomes.

Regarding age and grade, no significant difference was found in the level of programming resilience according to the measurement of the PRSUS. In Chisholm-Burns et al. (2019), academic resilience among pharmacy students was investigated. Their result also showed that the age difference in academic resilience was not significant. However, compared with students in higher grades, first year pharmacy students showed significantly higher levels of academic resilience. Conversely, the research of Lanuza et al. (2020) indicated that, when a student is older and they experience more academic difficulties, their level of academic resilience will be higher. One possible explanation for these inconsistent findings might have to do with the context in which the resilience arises. Regarding programming learning, the effect is never instant. Coding requires long-term learning and practice. When students learn more, they encounter more difficulties, and they are more likely to have higher levels of programming resilience if they can persevere. In this study, the level of programming resilience in fourth year students was higher than that of the students in the three lower grades, although the difference was not significant. Thus, if a wider range of grades was examined, a significant difference might result between students in higher and lower grades.

This research has some limitations that should be addressed in future studies. First, the participants were undergraduates majoring in science and engineering at universities in China. The scale based on these data is therefore limited in its universality. Further verification of the scale is needed to determine if it can also be applied to students in higher or lower grades and whether it is applicable to students with other majors. Second, the sample consisted of far fewer third and fourth year students than first and second year students, leaving students in the higher grades potentially underrepresented. Future research should ensure that students from all grades are evenly represented.

In future study, the scale will help researchers carry out deeper research on students' programming resilience, understand the current situation of students' programming resilience, explore the factors that influence programming resilience, and explore the interrelationship between various teaching factors including programming resilience. In addition, researchers can further explore the relationship between programming resilience and students' programming achievement, and how programming resilience affects programming achievement. Moreover, it may help researchers examine the influence of different teaching interventions on students' levels of programming resilience, which can help teachers improve their programming teaching, help students overcome the difficulties in the process of programming learning, and obtain successful learning outcomes.

References

- Abdunabi, R., Hbaci, I., & Ku, H. Y. (2019). Towards enhancing programming self-efficacy perceptions among undergraduate information systems students. *Journal of Information Technology and Education: Research*, 18, 185-206.
- Anthony, E. J., Anthony, E. I., & Koupernik, C. (1974). The syndrome of the psychologically invulnerable child. *The Child in His Family*, 3, 529-544.
- Aronson, J., Fried, C. B., & Good, C. (2002). Reducing the effects of stereotype threat on African American college students by shaping theories of intelligence. *Journal of Experimental Social Psychology*, 38(2), 113-125. <https://doi.org/10.1006/jesp.2001.1491>
- Autin, Frédérique, & Croizet, J. C. . (2012). Improving working memory efficiency by reframing metacognitive interpretation of task difficulty. *Journal of Experimental Psychology: General*, 141(4), 610-618. <https://doi.org/10.1037/a0027478>

- Balanskat, A., & Engelhardt, K. (2014). Computing our future - Priorities, school curricula and initiatives across Europe. *European Schoolnet*.
- Bennedsen, J., & Caspersen, M. E. (2019). Failure rates in introductory programming - 12 years later. *ACM Inroads*, 10(2), 30-35. <https://doi.org/10.1145/3324888>
- Blackwell, L. S., Trzesniewski, K. H., & Dweck, C. S. (2007). Implicit theories of intelligence predict achievement across an adolescent transition: A longitudinal study and an intervention. *Child Development*, 78(1), 246-263. <https://doi.org/10.1111/j.1467-8624.2007.00995.x>
- Cassidy, S. (2016). The Academic Resilience Scale (ARS-30): A new multidimensional construct measure. *Frontiers in Psychology*, 7(1787), 1-11. <https://doi.org/10.3389/fpsyg.2016.01787>
- Çetinkaya, A., & Baykan, Ö. K. (2020). Prediction of middle school students' programming talent using artificial neural networks. *Engineering Science and Technology*, 23(6), 1301-1307. <https://doi.org/10.1016/j.jestch.2020.07.005>
- Chapman, M. T., Lines, R. L. J., Crane, M., Ducker, K. J., Ntoumanis, N., Peeling, P., Parker, S. K., Quested, E., Temby, P., Thøgersen-Ntoumani, C., & Gucciardi, D. F. (2020). Team resilience: A scoping review of conceptual and empirical work. *Work and Stress*, 34(1), 57-81. <https://doi.org/10.1080/02678373.2018.1529064>
- Chisholm-Burns, M. A., Spivey, C. A., Sherwin, E., Williams, J., & Phelps, S. (2019). Development of an instrument to measure academic resilience among pharmacy students. *American Journal of Pharmaceutical Education*, 83(6), 1390-6896. <https://doi.org/10.5688/ajpe6896>
- Chmitorz, A., Kunzler, A., Helmreich, I., Tüscher, O., Kalisch, R., Kubiak, T., Wessa, M., & Lieb, K. (2018). Intervention studies to foster resilience – A systematic review and proposal for a resilience framework in future intervention studies. *Clinical Psychology Review*, 59, 78-100. <https://doi.org/10.1016/j.cpr.2017.11.002>
- Crandall, A., Miller, J. R., Cheung, A., Novilla, L. K., Glade, R., Novilla, M. L. B., Magnusson, B. M., Leavitt, B. L., Barnes, M. D., & Hanson, C. L. (2019). ACEs and counter-ACEs: How positive and negative childhood experiences influence adult health. *Child Abuse and Neglect*, 96, 1-9. <https://doi.org/10.1016/j.chiabu.2019.104089>
- Deci, E. L., Vallerand, R. J., Pelletier, L. G., & Ryan, R. M. (1991). Motivation and education: The self-determination perspective. *Educational Psychologist*, 26(3-4), 325-346.
- Durak, H. Y. (2018). Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences. *Journal of Computer Assisted Learning*, 34(6), 740-752. <https://doi.org/10.1111/jcal.12281>
- Elçiçek, M., & Karal, H. (2020). A framework proposal for the design of video-assisted online learning environments for programming teaching. *Elementary Education Online*, 19(3), 1820-1837. <https://doi.org/10.17051/ilkonline.2020.735175>
- Field, A. P. (2009). *Discovering statistics using SPSS*. SAGE Publications.
- Garnezy, N., Masten, A. S., & Tellegen, A. (1984). The study of stress and competence in children: A building block for developmental psychopathology. *Child Development*, 55(1), 97-111.
- George, D. (2003). *SPSS for Windows step by step: A simple guide and reference* (11th ed.). Prentice Hall.
- Hoge, E. A., Austin, E. D., & Pollack, M. H. (2007). Resilience: Research evidence and conceptual considerations for posttraumatic stress disorder. *Depression and Anxiety*, 24(2), 139-152. <https://doi.org/10.1002/da.20175>
- Hung, M. (2016). Teacher readiness for online learning: Scale development and teacher perceptions. *Computers & Education*, 94, 120-133. <https://doi.org/10.1016/j.compedu.2015.11.012>
- Jegade, P. O., Olajubu, E. A., Ejidokun, A. O., & Elesemoyo, I. O. (2019). Concept-based analysis of java programming errors among low, average and high achieving novice programmers. *Journal of Information Technology Education: Innovations in Practice*, 18, 49-59.
- Johnston-Wilder, S., Lee, C., Garton, L., Goodlad, S., & Brindley, J. (2013, November 18-20). *Developing coaches for mathematical resilience* [Paper presentation]. 6th International Conference of Education, Research and Innovation. Seville, Spain.
- Keller, L., & John, I. (2020). Motivating female students for computer science by means of robot workshops. *International Journal of Engineering Pedagogy*, 10(1), 94-108. <https://doi.org/10.3991/ijep.v10i1.11661>
- Kim, S., You, K., Hong, K., & Cho, Y. (2019). The analysis of resilience of programming class' students for basic liberal arts. *Journal of the Korea Institute of Information and Communication Engineering*, 23(7), 801-806.

- Kline, R. B. (2011). *Principles and practice of structural equation modeling*. Guilford Press.
- Kooken, J., Welsh, M. E., McCoach, D. B., Johnston-Wilder, S., & Lee, C. (2016). Development and validation of the mathematical resilience scale. *Measurement and Evaluation in Counseling and Development*, 49(3), 217-242. <https://doi.org/10.1177/0748175615596782>
- Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science: A practical primer for t-tests and ANOVAs. *Frontiers in Psychology*, 4(11), 1-12. <https://doi.org/10.3389/fpsyg.2013.00863>
- Lanuza, M. H., Rizal, R. A. G., Aligam, N. P., & Uy, R. (2020). Contextualize program of strengthening academic resilience level of the secondary education students. *Journal of Critical Reviews*, 7(11), 286-292. <https://doi.org/10.31838/jcr.07.11.46>
- Lindberg, R. S. N., Laine, T. H. , & Lassi, H. . (2019). Gamifying programming education in k-12: A review of programming curricula in seven countries and programming games. *British Journal of Educational Technology*, 50(4), 1979-1995.
- Lock, S., Rees, C. S., & Heritage, B. (2019). Development and validation of a brief measure of psychological resilience: The state–trait assessment of resilience scale. *Australian Psychologist*, 55(1), 10-25. <https://doi.org/10.1111/ap.12434>
- Luthar, S. S. (2006). Resilience in development: A synthesis of research across five decades. In Dante, C., & Donald, J. C (Eds.), *Developmental psychopathology* (pp. 739-795). John Wiley & Sons. <https://doi.org/10.1002/9780470939406.ch20>
- Malik, S. I., & Coldwell-Neilson, J. (2018). Gender differences in an introductory programming course: New teaching approach, students' learning outcomes, and perceptions. *Education and Information Technologies*, 23(6), 2453-2475. <https://doi.org/10.1007/s10639-018-9725-3>
- Margulieux, L. E., Morrison, B. B., & Decker, A. (2020). Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. *International Journal of STEM Education*, 7(1), 416-427. <https://doi.org/10.1186/s40594-020-00222-7>
- Martin, A. J. (2001). The Student Motivation Scale: A tool for measuring and enhancing motivation. *Australian Journal of Guidance & Counselling*, 11, 1-20.
- Martin, A. J. (2002). Motivation and academic resilience: Developing a model for student enhancement. *Australian Journal of Education*, 46(1), 34-49. <https://doi.org/10.1177/000494410204600104>
- Martin, A. J. (2003). The Student Motivation Scale: Further testing of an instrument that measures school students' motivation. *Australian Journal of Education*, 47(1), 88-106. <https://doi.org/10.1177/000494410304700107>
- Martin, A. J., & Marsh, H. W. (2006). Academic resilience and its psychological and educational correlates: A construct validity approach. *Psychology in the Schools*, 43(3), 267-281. <https://doi.org/10.1002/pits.20149>
- Martin, A. J., & Marsh, H. W. (2008). Academic buoyancy: Towards an understanding of students' everyday academic resilience. *Journal of School Psychology*, 46(1), 53-83. <https://doi.org/10.1016/j.jsp.2007.01.002>
- Netemeyer, R. G., Bearden, W. O., & Sharma, S. (2003). *Scaling procedures: Issues and applications*. Sage.
- Oyserman, D., Elmore, K., Novin, S., Fisher, O., & Smith, G. C. (2018). Guiding people to interpret their experienced difficulty as importance highlights their academic possibilities and improves their academic performance. *Frontiers in Psychology*, 9, 1-15. <https://doi.org/10.3389/fpsyg.2018.00781>
- Pattanaphanchai, J. (2019). An investigation of students' learning achievement and perception using flipped classroom in an introductory programming course: A case study of Thailand higher education. *Journal of University Teaching and Learning Practice*, 16(5), 1-19.
- Ricketts, S. N., Engelhard, G., & Chang, M. (2017). Development and validation of a scale to measure academic resilience in mathematics. *European Journal of Psychological Assessment*, 33(2), 79-86. <https://doi.org/10.1027/1015-5759/a000274>
- Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1), 68-78. <https://doi.org/10.1037/0003-066X.55.1.68>
- Saito, T., & Watanobe, Y. (2020). Learning path recommendation system for programming education based on neural networks. *International Journal of Distance Education Technologies*, 18(1), 36-64. <https://doi.org/10.4018/IJDET.2020010103>
- Sisto, A., Vicinanza, F., Campanozzi, L. L., Ricci, G., Tartaglini, D., & Tambone, V. (2019). Towards a transversal definition of psychological resilience: A literature review. *Medicina (Lithuania)*, 55(11). <https://doi.org/10.3390/medicina55110745>

- Trigueros, R., Magaz-González, A. M., García-Tascón, M., Alias, A., & Aguilar-Parra, J. M. (2020). Validation and adaptation of the academic-resilience scale in the Spanish context. *International Journal of Environmental Research and Public Health*, 17(11), 1-11. <https://doi.org/10.3390/ijerph17113779>
- Tsai, C. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Computers in Human Behavior*, 95, 224-232. <https://doi.org/10.1016/j.chb.2018.11.038>
- Ungar, M. (2006). Resilience across cultures. *British Journal of Social Work*, 38(2), 218-235. <https://doi.org/10.1093/bjsw/bcl343>
- Ungar, M. (2011). The social ecology of resilience: Addressing contextual and cultural ambiguity of a nascent construct. *American Journal of Orthopsychiatry*, 81(1), 1-17. <https://doi.org/10.1111/j.1939-0025.2010.01067.x>
- Ungar, M. (2013). Resilience, trauma, context, and culture. *Trauma, Violence, and Abuse*, 14(3), 255-266. <https://doi.org/10.1177/1524838013487805>
- Vallerand, R. J., & Blssonnette, R. (1992). Intrinsic, extrinsic, and amotivational styles as predictors of behavior: A prospective study. *Journal of Personality*, 60(3), 599-620.
- Wagnild, G. M., & Young, H. M. (1993). Development and psychometric evaluation of the Resilience Scale. *Journal of Nursing Measurement*, 1(2), 165-178.
- Waite, J., Curzon, P., Marsh, W., & Sentance, S. (2020). Difficulties with design: The challenges of teaching design in K-5 programming. *Computers & Education*, 150(6), 1-28. <https://doi.org/10.1016/j.compedu.2020.103838>
- Yarygina, O. (2020, January 29-30). *Learning analytics of CS0 students programming errors: The case of data science minor* [Paper presentation]. ACM International Conference Proceeding Series (pp. 149-152). New York, NY. <https://doi.org/10.1145/3377290.3377319>
- Yeager, D. S., & Dweck, C. S. (2012). Mindsets that promote resilience: When students believe that personal characteristics can be developed. *Educational Psychologist*, 47(4), 302-314. <https://doi.org/10.1080/00461520.2012.722805>
- Ying, C., Baoyong, L., & Yajuan, Z. (2010). Relationship among stress of entrance examination, resilience and stress distress of middle school students. *Chinese Journal of Clinical Psychology*, 18(02), 180-182.
- Zhang, Y., Chen, T., & Wang, C. (2020). Factors influencing students' willingness to choose blended learning in higher education. *Lecture Notes in Computer Science*, 12218, 289-302. https://doi.org/10.1007/978-3-030-51968-1_24
- Zhen, R., Liu, R. D., Wang, M. T., Ding, Y., Jiang, R., Fu, X., & Sun, Y. (2020). Trajectory patterns of academic engagement among elementary school students: The implicit theory of intelligence and academic self-efficacy matters. *British Journal of Educational Psychology*, 90(3), 618-634. <https://doi.org/10.1111/bjep.12320>

Corresponding author: Dong Yan , yan.dong@bnu.edu.cn

Copyright: Articles published in the Australasian Journal of Educational Technology (AJET) are available under Creative Commons Attribution Non-Commercial No Derivatives Licence ([CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)). Authors retain copyright in their work and grant AJET right of first publication under CC BY-NC-ND 4.0.

Please cite as: Fu, Q., Zhang, L-W., Hong, J-C., & Dong, Y. (2021). Development and verification of the programming resilience scale for university students. *Australasian Journal of Educational Technology*, 37(6), 141-155. <https://doi.org/10.14742/ajet.6833>

Appendix A Programming Resilience Scale for university students (English version)

Persistence
<p>When I encounter difficulties in programming that might cause most people to give up, I will keep trying.</p> <p>I stand strong and keep fighting even if I cannot get help and support for a long time due to problems in programming.</p> <p>I am not going to lower my standards or change my goals simply because I encounter some difficulties in programming.</p>
Perceived value
<p>Having a solid knowledge of programming helps me understand more-complex issues in other disciplines.</p> <p>Programming thinking can help me deal with things that matter to me.</p> <p>A knowledge of programming is very helpful no matter what subject I decide to study.</p>
Difficulty cognition
<p>Good programmers experience difficulties when solving programming problems.</p> <p>People in my peer group struggle sometimes with programming.</p> <p>Having difficulty in programming learning or having repeated errors in the program is a normal part of working on programming.</p>
Incremental belief
<p>Anyone can achieve satisfactory results on a programming exam by working hard.</p> <p>Anyone can master the basic concepts of programming through hard work.</p> <p>Anyone can master programming skills through hard work.</p>