AMERICAN PSYCHOLOGICAL ASSOCIATION

Translational Issues in Psychological Science

© 2022 American Psychological Association
ISSN: 2332-2179

2022, Vol. 8, No. 1, 104–124
https://doi.org/10.1037/tps0000293

Predictive Effects of Creative Abilities and Attitudes on Performance in University-Level Computer Science Courses

Richard J. Daker[1], Griffin A. Colaizzi[1], Ariana M. Mastrogiannis[1], Micah Sherr[2], Ian M. Lyons[1], and Adam E. Green[1]

[1] Department of Psychology, Georgetown University
[2] Department of Computer Science, Georgetown University

Much of what determines success in computer science involves applying previously learned knowledge in new and creative ways. However, while prior work has examined several cognitive and affective factors that predict success in computer science courses, no work we are aware of has assessed whether measures of creativity predict computer science achievement. To begin to fill this important gap, we collected several measures of ability in and attitudes toward creativity and asked which creativity measures, if any, most strongly predicted success in computer science courses. We found that individual differences in analogical reasoning performance when students were prompted to "be creative" predicted grades earned in computer science courses, as did individual differences in the extent to which students view computer science as "creative." These two variables were each predictive even when controlling for one another, demonstrating that analogical creativity and views about the relevance of creativity in computer science predict unique variance in computer science grades. Moreover, these variables also remained predictive even when controlling for several other creativity measures, including the commonly used Alternative Uses Task and Remote Associates Test, which did not predict individual differences in computer science achievement. These results suggest that the ability to make connections between seemingly unrelated concepts may aid computer science achievement. Results also suggest that students who approach computer science as a creative endeavor may find more success in their courses. These findings suggest the promise of future intervention-based approaches that foster analogical creativity and highlight the creativity involved in computer science.

What is the significance of this article for the general public?
Many computer scientists consider computer science to be a creative discipline, but no research we are aware of has tested whether creative abilities or attitudes are predictive of success in university-level computer science courses. Here, we found that students who were high in their ability to identify creative analogies and students who saw computer science as more "creative" did better on their computer science exams than students who were low on these measures. While this study was correlational, future work based on these findings could test whether improving students' creative analogical reasoning abilities or making students see computer science as a creative exercise would boost performance in university computer science courses.

Keywords: creativity, computer science, STEM, analogical reasoning, education

Richard J. Daker https://orcid.org/0000-0002-7416-4791
Griffin A. Colaizzi https://orcid.org/0000-0003-1866-8335
Richard J. Daker and Griffin A. Colaizzi share first authorship.

This research was supported by grants to Adam E. Green from the National Science Foundation (DRL-1420481, EHR-1661065, EHR-1920682). We thank Addison Woods and Jeremy Bolton for allowing us to collect data from their introductory computer science students, and we would like to thank all participants for their contribution to this project.

Correspondence concerning this article should be addressed to Richard J. Daker, Department of Psychology, Georgetown University, 306N White-Gravenor Hall, 37th and O Streets, N.W., Washington, DC 20057, United States. Email: rjd107@georgetown.edu

With the pervasiveness of computers in daily life, there has been a rapid increase in demand for jobs in computer science (Bureau of Labor Statistics, U.S. Department of Labor, n.d.). Indeed, companies consistently rate computer science skills among the most sought-after skills for employment (Petrone, 2019). Despite this large demand, many of the available computer science positions remain unfilled (Pfeiffer, 2019). The dearth of experienced workers to fill these positions is alarming, as innovation in computer science drives progress in industry and society. Innovation in computer science is responsible for the growth and change of industries including in earth science, political science, social sciences, economics, and business (Yang et al., 2017). For students to be eligible to fill these jobs, they must have gained the necessary expertise, which is generally demonstrated through success in university-level computer science courses (Aasheim et al., 2019). The identification of factors that predict or impede success in computer science classes could be used to inform interventions that aim to assist students in gaining expertise in computer science.

Computer Science and Creative Abilities

Previous work has shown that ability in various cognitive domains is associated with success in computer science courses, including mathematical ability (Bergin & Reilly, 2005; Byrne & Lyons, 2001; Evans & Simkin, 1989; Konvalina et al., 1983; Leeper & Silver, 1982; Werth, 1986; Wilson & Shrock, 2001), spatial reasoning ability (Adelson, 1981; Cooke, & Schvaneveldt, 1988; McKeithen et al., 1981), and abstract reasoning ability (Kurtz, 1980). While many types of cognitive ability have been shown to predict computer science success, to our knowledge, no previous work has assessed whether creative abilities are predictive of computer science achievement. This is an important gap in the literature, because there are strong reasons to expect that creative abilities —commonly defined in the creativity literature as cognition that results in the generation of novel and useful output (Runco & Jaeger, 2012)—would support success in computer science courses. Perhaps most notably, computer scientists themselves view the field as a creative one in which creativity is "demanded and encouraged" (Knobelsdorf & Romeike, 2008) and is identified as one of the "core soft skills" of computer science (Romeike, 2007a). Indeed, there are many aspects of computer science that require creativity to be successful. Researchers have argued that creativity is needed for effective software design and construction (Gu & Tong, 2004), to create efficient algorithms (Scragg et al., 1994), and to successfully design in many domains of computer science including graphics, AI, and information systems (Cennamo et al., 2011; Saunders & Thagard, 2005).

Creativity is likely to aid performance in the computer science classroom as well. In many computer science classes, students are taught several basic concepts and then assessed on their ability to use and combine these basic building blocks to solve a new problem that they have not yet encountered. For example, a typical data structures course—a foundational course in a typical undergraduate computer science program—has students learn a variety of methods to program a computer to maintain and quickly access information. That, in and of itself, does not solve modern computer programming problems. However, these simple concepts can be combined in creative ways to complete much more difficult tasks (i.e., anything from searching online documents for key phrases to constructing a new cryptocurrency). Almost all large computer programs are complex combinations of relatively simple concepts, and the exercise of programming requires the programmer to determine which simple components should be used and where and how they should be applied. Consequently, computer science courses assess students not only on their mastery of specific concepts, but also on their ability to problem-solve tasks that require clever composition of those concepts. Taking existing information and applying it to solve problems in new situations requires creativity, and creativity may help students succeed in computer science courses. Past work has also developed interventions that aim to improve computer science grades by promoting creativity (Apiola et al., 2010; Cennamo et al., 2011; Peteranetz et al., 2017; Peteranetz, Flanigan, et al., 2018, Peteranetz, Wang, et al., 2018), but no work to our knowledge has assessed whether individual differences related to creativity are predictive of computer science achievement.

There are several types of creative ability that may aid students in computer science courses. Aspects of computer science, like algorithm creation and implementation, design processes, and software development are inherently creative and have been considered as complex problem-solving activities requiring creative engagement (Glass, 2006; Hill, 1998; Scragg et al., 1994). Creative idea generation, or the ability to produce novel and useful (i.e., creative) ideas, is widely accepted as a key component in the creative problem solving and innovation process (Amabile, 1988; Baer, 2012). Creative idea generation ability could be important for success in computer science courses. Another type of creativity that may support performance in computer science courses is creative insight. Creative insight refers to situations in which the correct answer to a problem just "pops into your head," and is often described as involving a "Eureka" moment or an "Aha" moment (Csikszentmihalyi & Sawyer, 2014; Kounios & Beeman, 2015). Students' creative insight abilities may allow them to arrive at solutions to new problems posed to them in their computer science exams, which often involve recognizing ways in which a previously learned concept could be applied to a new situation. In their chapter entitled "Creativity in Computer Science," Saunders and Thagard (2005) argue that creative analogical reasoning, or the ability to make novel connections between distant concepts, might also be important for success in computer science. The ability to understand the relationships between core computer science concepts and utilize these connections to solve real-world problems is a crucial skill for the effective computer science student (Scragg et al., 1994). This ability might affect computer science success as those who can make creative connections across different computer science concepts might have an easier time utilizing concepts together to solve difficult computational problems. Interestingly, performance on creative analogical reasoning tasks has been shown to improve with a simple cue to "think creatively" as participants complete the task (Weinberger et al., 2016). This and other similar effects demonstrate that individuals can alter their "creative state," allowing them to approach a problem in a creative or uncreative way (Green, 2016). The ability to consciously augment one's creative state may be another form of creative ability that supports success in computer science by allowing students to dynamically shift their approach as they attempt to generate solutions to difficult problems.

Computer Science and Attitudes toward Creativity

While different types of creative ability are likely to support success in computer science, there are noncognitive factors that might contribute to success in computer science classes as well. Many studies have illustrated numerous noncognitive factors that predict success in computer science classes, including attitudes toward computer science and self-efficacy in computer science (Charlton & Birkett, 1999; Wiedenbeck, 2005; Wiedenbeck et al., 2004). Moreover, Wilson and Shrock (2001) illustrated that an important predictor of success in an introductory computer science course was comfort level, operationalized by the extent to which a student felt anxiety about the course. While anxiety about computer science is likely to affect success in computer science courses, anxiety toward creativity may also play a role in shaping computer science achievement. Creativity anxiety, an anxiety specific to creative thinking, has recently been identified as an anxiety existing across diverse content domains which predicts individual differences in creative achievement (Daker et al., 2020; Ren et al., 2021). Given that computer science is a creative discipline (Glass, 2006; Leach & Ayers, 2005; Scragg et al., 1994), creativity anxiety may affect students' performance in computer science courses.

Additionally, student perceptions about the extent to which computer science is creative may also predict success in computer science classes. Previous work has shown that computer science students perceive some aspects of software development as more creative than others and preferred working on the aspects they viewed as more creative (Gu & Tong, 2004; Romeike, 2006). Understanding the extent to which students perceive the involvement of creativity in computer science could provide insight into how professors should be teaching and talking about computer science. If individual differences in the perceived "creativeness" of computer science predicts success in computer science courses, emphasizing the creative nature of computer science might increase interest and enjoyment in the subject or lead to better computer science outcomes. These perceptions of the overlap between computer science and creativity may also moderate the effects of students' creative ability and/or anxiety. Indeed, those who have different beliefs about the role of creativity in computer science might approach computer science problems in different ways or with different solutions. For example, a student who does not believe computer science involves creativity might be less likely to benefit from her creative ability by ignoring or not applying creative solutions to a computer science problem. Furthermore, students with high levels of creativity anxiety who do not see computer science as involving creativity might not experience the possible deleterious effects of creativity anxiety.

Levels of Computer Science Expertise as a Potential Moderator of the Predictive Effects of Creativity on Computer Science Achievement

Finally, it is possible that measures of ability in, anxiety about, and perceived involvement of creativity in computer science might have differential impacts on computer science success at different course levels. It is possible that certain creative abilities only impact students' performance in lower level courses, or that creativity anxiety more strongly impacts students in upper level courses. For example, creative idea generation might be more important in advanced computer science courses because students have more freedom and are expected to creatively work with and combine different core concepts of computer science. Identifying differences in what predicts success in computer science courses at different levels of expertise could point to different types of interventions that should be pursued in an effort to boost achievement.

The Present Study

In the present study we recruited students in both introductory and advanced university-level computer science courses to complete several measures of creative ability and attitudes at the start of the semester. With permission from students and their instructors, we then obtained grade reports for each student as a measure of their success in the course at the end of the semester. This allowed us to explore the extent to which creative abilities and attitudes predicted real-world computer science outcomes. In addition to assessing zero-order associations between measures of creative abilities and attitudes and computer science course achievement, we also collected several control measures (verbal reasoning, nonverbal reasoning, general anxiety, and computer science anxiety). This allowed us to assess whether any observed associations between measures of creativity and computer science could be explained by noncreative factors; increasing the specificity with which we could link measures of creativity with computer science performance. While the main goal of this study was exploratory in nature (i.e., to explore whether different creativity measures are predictive of computer science grades), we also tested the hypotheses that associations between measures of creativity and computer science performance would be moderated by the perceived involvement of creativity in computer science or by whether students were enrolled in an introductory course or an advanced course.

Method

Participants

A total of 153 students participated in the study. Participants were undergraduate and graduate students enrolled in specific computer science courses at Georgetown University. Students were recruited from two different courses: Computer Science 1 (listed as COSC 051), an introductory course that is the first course computer science majors take in their major (hereon referred to as the "intro-level course"), and Intro to Network Security (listed as COSC 435), a course for advanced undergraduates and graduate students taught by Micah Sherr. After removing participants who failed two or more attention checks (13 students) or who eventually dropped the computer science course (eight students), 132 participants remained. Sixty-nine students in the final analytic sample ($M_{\text{age}} = 19.49$, 36 female) were from the intro-level course and were recruited from three sections of the course in the fall of 2018. The final analytic sample included a total of 63 students from the upper-level course ($M_{\text{age}} = 22.38$, 24 female). Because fewer students take the upper-level course each year, participants for the upper-level course were recruited in both the fall of 2018 (35 students) and the fall of 2019 (28 students). Students participated for extra credit in their computer science course. Informed electronic consent was obtained for all participants before the study in accordance with the guidelines established by the university's Institutional Review Board.

Procedure

Within the first month of their computer science courses, participants completed an online battery of survey and ability measures (discussed in further detail below). Participants were encouraged to take breaks in between measures (before each task or questionnaire began, participants saw a screen prompting them to take a break if they would like to do so), but were instructed not to take breaks while in the middle of a questionnaire or task. All survey and ability measures were presented in a randomized order. The full battery of tasks and questionnaires took participants, on average, 35 min to complete. The instructor of each class provided the grades participating students earned on assignments and exams throughout the semester.

Measures

Ability Measures

Alternative Uses Task. Participants' ability to generate new and unique ideas was measured using the Alternative Uses Task (AUT) developed by Guilford (Guilford, 1967). To complete this task, participants were given the following instructions: "For this section, list as many alternative uses for a **brick** as you can think of in two minutes." Participants typed their responses. Responses were scored by two independent raters for originality on a 1 (*very obvious and ordinary use*) to a 5 (*very imaginative, recontextualized use*) scale. Raters were asked to consider how uncommon, remote, and clever each response was while determining a score. A final originality score (referred to as AUT Originality) was calculated by averaging the originality scores of the two raters. The same two independent raters also scored responses for fluency on a 0 (*inappropriate, irrelevant response*) to 1 (*appropriate, legitimate response*) scale. A final fluency score (referred to as AUT Fluency) was calculated by summing the fluency score for each response; this score reflects the number of unique ideas a participant generated while completing the task. Interrater reliability was high for both originality and fluency (intraclass correlation coefficients: originality: .829; fluency: .997). In the creativity literature, the AUT is often used as a measure of creative divergent thinking ability, or the ability to think in new and unconstrained ways, though convergent thinking is also necessary to complete the task (e.g., Cortes et al., 2019).

Remote Associates Task. Participants' ability to arrive at correct answers to problems through creative insight was measured using Mednick's Remotes Associate Task (RAT; Mednick, 1962). In a trial of this task, participants are given three stimulus words and are told to generate a fourth word that would create a common phrase when combined with the stimulus words. For example, participants were presented with a word triad such as [show/life/row] and generated a fourth word, in this case, [boat], which makes three possible compound words/phrases with the original triad: showboat, lifeboat, and rowboat. Participants had 10 s for each trial to type their response. Participants completed a total of 15 trials that had an average solution rate of 49.2% (with a range of solution rates from 29% to 79%) in a norming study done by Bowden & Jung-Beeman (2003). The stimuli participants responded to for this task can be viewed in the Appendix. Cronbach's α for this measure was .82. The RAT is a commonly used measure in the creativity literature and is standardly interpreted as a measure of convergent creative thinking ability, though it involves divergent thinking as well (Cortes et al., 2019).

Analogy Finding Task. Creative analogical reasoning was tested using the Analogy Finding Task (Green et al., 2017; Weinberger et al., 2016). In this task, participants are shown a matrix of word-pairs, with five word-pairs shown as a column on the left side of the screen (stem pairs) and 20 word-pairs shown as a row at the top of the screen (completion pairs). The instructions participants received are as follows: "Your task is to make analogies by combining word-pairs on the left side of the grid with word-pairs along the top of the grid. Each word-pair should be read as '[Top Word] is to [Bottom Word].' For example, 'Helmet is to Head.' Check the boxes to indicate when a word-pair from the top combines with a word-pair on the left to make a valid analogy. Try to make as many analogies as you can. However, only valid analogies should be listed, so do not list analogies unless you can describe how the two word pairs are analogous." Each stem pair could be combined with three or four of the completion pairs to form valid analogies. A total of 17 valid analogies can be found in each matrix participants completed (out of a possible 100 possible combinations of stem pairs and completion pairs). The valid analogies ranged in their semantic distance, an increasingly commonly used index of creativity that refers to how likely words are to cooccur near each other in text corpuses (Beaty & Johnson, 2020; Prabhakaran et al., 2014). For example, the valid analogy "[Kitten] is to [Cat] as [Puppy] is to [Dog]" is low in semantic distance and is considered a relatively uncreative analogy and "[Kitten] is to [Cat] as [Spark] is to [Fire]" is high in semantic distance and is considered a relatively creative analogy. The stimuli participants responded to for this task can be viewed in the Appendix.

After completing one analogy matrix following the instructions above, participants we presented with another analogy matrix, this time with the following additional instructions: "This time, please think creatively as you search for valid analogies. Some analogies may not be obvious right away, so be sure to look for abstract connections." Previous work has shown that this "creativity cue" increases the amount of valid analogies participants select without increasing the number of invalid analogies chosen (Green et al., 2017; Weinberger et al., 2016). Two versions of the Analogy Finding matrices were used, and the order in which they were presented was counterbalanced. Note that one study in Weinberger et al. (2016). presented both matrix versions without the creativity cue and found no evidence of practice effects, suggesting that differences in performance on the cued and uncued matrices can be attributed to the presence of the creativity cue. Separate reliability estimates were obtained for each matrix version (Matrix 1, Matrix 2) and cue (cued, uncued) combination: Matrix 1, cued Cronbach's $\alpha = .87$; Matrix 1, uncued Cronbach's $\alpha = .84$; Matrix 2, cued Cronbach's $\alpha = .84$; Matrix 2, uncued Cronbach's $\alpha = .85$.

Two separate dependent variables were produced by this task—the number of correct analogies selected on the uncued matrix (Analogy Finding Task—Uncued) and the number of correct analogies selected on the matrix with the creativity cue (Analogy Finding Task—Creativity Cue), which provide a measure of standard analogical reasoning and creative analogical reasoning, respectively. To penalize incorrect answers, the number of correct responses on a matrix was residualized with respect to the number of incorrect responses; thereby, effectively controlling for the number of incorrect responses participants gave. Note that when both of these measures are present in the same regression model, the coefficient of the Analogy Finding Task—Creativity Cue can be interpreted as the effect of the creativity cue, or one's ability to consciously augment their creative state during analogical reasoning.

Ravens Advanced Progressive Matrices. Participants were administered a portion of the Ravens Advanced Progressive Matrices (Ravens; Raven et al., 1998). This task requires participants to complete a visual pattern in which one section of nine is missing. Participants answer by selecting one of eight options to fill in the missing section. One half of the Ravens questions (all even numbered questions) were selected for this portion of the experiment. After completing a practice problem, participants were given 10 min to complete as many of the selected questions as possible (17 total). Scores were calculated by summing the total correct responses. Cronbach's α for this measure was .69. This task was included primarily as a covariate to allow us to control for nonverbal reasoning ability.

Syllogistic Reasoning Task. Participants completed a variety of three term syllogistic reasoning problems with different model structures that use spatial and nonspatial language. This task, known as the Multidimensional Relational Reasoning Task (Cortes et al., 2021), involves relations along multiple dimensions to make the problems more difficult than syllogistic reasoning tasks that only involve relations along a single dimension. Participants see two premises and must determine whether a certain conclusion is true or false. For example: Premise 1: Matthew is above and to the left of Andrew. Premise 2: Thomas is above and to the right of Andrew. Conclusion: THOMAS IS ABOVE AND TO THE RIGHT OF MATTHEW. The correct answer here is "true." Some trials feature nonspatial language in which the comparative features "above," "below," "left," and "right" are replaced with "more/less certain" and "more/less excited." Participants were presented with 15 different trials and were given 15 s to answer each trial. Scores were calculated by summing the number of correct responses. Cronbach's α for this measure was .46; given that reliability for this task was low, associations between performance on this task and other measures should be interpreted with caution. This task (later referred to as "Syllogisms") was included primarily as a covariate to allow us to control for verbal reasoning ability. The stimuli participants responded to for this task can be viewed in the Appendix.

Survey Measures

Creativity Anxiety Scale. Participants completed the Creativity Anxiety Scale (CAS; Daker et al., 2020). The CAS consists of two item types, creativity anxiety items (CA) and noncreativity anxiety control items (NAC). For all items, participants are asked to indicate how anxious each situation would make them. CA items measure anxiety toward situations that require being creative (e.g., "Having to come up with a unique way of doing something" and "Having to think in an open-ended and creative way"). NAC items measure anxiety toward very similar situations as those presented in the CA items but that remove the need to be creative (e.g., "Having to precisely follow an established method of doing something" and "Having to think in a precise and methodical way"). This allows for anxiety toward the noncreative demands of the situations presented in the CA items to be measured and controlled for. Participants provided anxiety ratings for each of the 16 items (8 CA, 8 NAC) by indicating how anxious they would be in the presented situations on a scale from 0 (*none at all*) to 4 (*very much*). Both CA and NAC scores range from 0 to 32, where higher scores indicate greater anxiety. Cronbach's α was .91 for CA items and .89 for NAC items. The CAS was included to provide a measure of anxiety specific to creative thinking.

Computer Science Anxiety. Participants responded to the single item "In general, how anxious about Computer Science are you?" on a scale from 1 (*not at all anxious*) to 10 (*very anxious*). This was meant to provide an indication of how anxious participants were about computer science. Note that while there are limitations to single-item measures, in other domains, such as math anxiety, single-item measures have performed comparably with longer scales in predicting outcomes (Núñez-Peña et al., 2014). This measure was included to assess whether any observed associations between other anxiety measures and computer science grades would hold even when controlling for anxiety about computer science itself.

Trait Anxiety Inventory. Participants completed the Trait Anxiety Inventory, a subscale of the State–Trait Anxiety Inventory (Spielberger et al., 1970). Participants were presented with a number of statements that describe how one generally feels (e.g., "I feel that difficulties are piling up so that I cannot overcome them" and "I feel pleasant" [reverse scored]) and asked to indicate how true that statement is of them on a scale from 1 (*almost never*) to 4 (*almost always*). This 20-item scale ranges in scores from 20 to 80, where higher scores indicate greater anxiety. Cronbach's α for this measure was .89. This scale was included to allow us to control for general trait anxiety when assessing relations between other anxiety measures and computer science grades.

Perceived Computer Science—Creativity Overlap. To assess perceived overlap between computer science and creativity, participants completed a modified version of the commonly used overlapping circles task (Aron et al., 1992; Necka et al., 2015). For this single-item measure, participants were asked "How much does Computer Science overlap with Creativity?" To indicate the degree to which they perceived "computer science" as overlapping with "creativity," participants were shown a series of seven Venn-diagrams that ranged from no overlap to almost complete overlap and asked to select the Venn-diagram that best represented the degree of overlap between Computer Science and Creativity. This provides a measure of the extent to which participants perceive computer science as involving creativity. Scores ranged from 1 to 7 with higher scores indicating a greater degree of perceived overlap. The measure of Perceived Computer Science—Creativity Overlap can be viewed in the Appendix. Note that for brevity, we hereon refer to this measure as "CompSci-Creativity Overlap."

Computer Science Outcome Measure

Grades. At the end of the semester, the research team received the grades students earned on assignments and exams throughout the semester. The dependent variable was the grades students earned on exams. This decision was made because the instructors of the courses indicated that differences in exam grades contributed the vast majority of variance in overall grades and because students often receive assistance with homework assignments (including working through the assignments with the aid of university-provided tutoring), whereas all exams were completed independently. The introductory courses had two exams and the upper-level courses had three exams, and the scores out of 100 were averaged for each participant to result in an average exam grade measure. This exam average was then squared to better approximate a normal distribution. All analyses were calculated using this squared exam average as our measure of performance in the computer science class. Note that using the untransformed exam data does not change any of the inferences afforded by the analyses to follow.

We also created a "Course Level" dummy variable that indicated whether each participant was enrolled in an intro-level course (0) or an upper-level course (1). This allowed us to control for course level when conducting analyses, and also allowed us to assess whether different measures are more predictive of computer science success at different stages of expertise with computer science.

Results

Do Measures of Creative Ability Predict Computer Science Grades?

We first sought to assess the extent to which measures of various types of creative ability (AUT Fluency, AUT Originality, RAT, and Analogy Finding Task—Creativity Cue) were associated with grades earned in computer science classes. Note that we additionally included more domain-general reasoning measures, like Ravens and Syllogisms, to assess whether any observed associations between creative abilities and Computer Science Grades would hold even when controlling for measures of noncreative reasoning. For descriptive statistics of all ability and academic measures, see Table 1. Figure 1 shows a zero-order Pearson correlation matrix that shows associations between all ability measures, Computer Science Grades, and Course Level. Results demonstrate that of all the ability measures included, only performance on the creativity cued Analogy Finding Task was significantly associated with Computer Science Grades ($r(130) = .224, p = .009$).

We next sought to test whether performance on the creativity-cued Analogy Finding Task would continue to predict unique variance in Computer Science Grades even when controlling for the other creative ability measures, for measures of noncreative reasoning, and for Course Level. We entered all variables shown in Figure 1 into a multiple regression model predicting Computer Science Grades. Results in Table 2 show that performance on the analogy matrix task when participants are given a cue to think creatively significantly predicted computer science grades ($\beta = .262$, $t(123) = 1.98$, $p = .050$), even when conservatively covarying all other cognitive ability measures (including performance on the uncued analogy matrix; though note that the p value associated with this unique association is exactly at the .05 level). We also note that the p value of the overall regression model is above .05 (model $F(8, 123) = 1.40$, $p = .203$, Overall adjusted $R^2 = .024$), which suggests that this overall model is not particularly useful at predicting Computer Science Grades. However, here we knowingly included in this model several measures that we had just shown (in Figure 1) were not predictive of Computer Science Grades, thereby penalizing the degrees of freedom for the overall regression model, because the key question we wished to address was whether performance on the creativity-cued Analogy Finding Task would continue to predict variance in Computer Science Grades even when stringently controlling for several other creativity and reasoning measures (we were less concerned with testing how well this collection of variables in aggregate could predict the outcome in question). As a result, while the overall model is not significantly predictive of Computer Science Grades after penalizing the degrees of freedom for each additional covariate, these results do provide useful evidence that the observed bivariate association between performance on the creativity-cued Analogy Matrix Task and

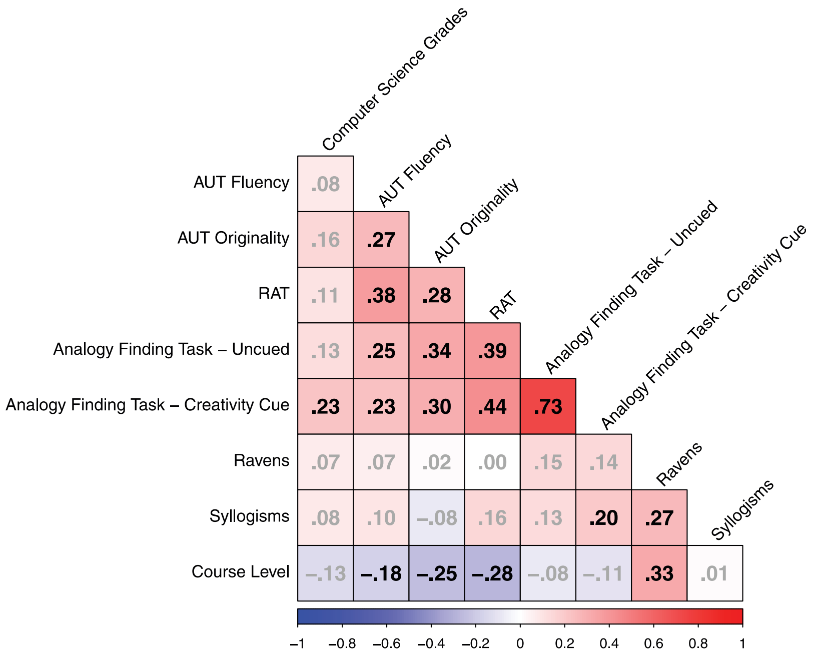
Table 1
Descriptive Statistics of Ability and Academic Measures

Measure	*M* (*SD*)	Range
Computer Science Grades	7,643.49 (1,295.23)	3,286.73–9,604
Course Level	N/A	0–1
AUT Fluency	9.09 (4.28)	0–21
AUT Originality	2.11 (0.56)	0–3.5
RAT	4.97 (3.71)	0–13
Analogy Finding Task—Uncued	0 (4.04)	−8.2–7.17
Analogy Finding Task—Creativity Cue	0 (3.97)	−10.5–8.27
Ravens	10.44 (2.89)	4–16
Syllogisms	7.36 (2.53)	2–14

Note. AUT = Alternative Uses Task; RAT = Remotes Associate Task. Table 1 shows descriptive statistics of all ability and academic measures. Note that for analysis purposes the measure of Computer Science Grades is squared (see Method). The descriptive statistics for the untransformed Computer Science Grades measure are as follows: *M* (*SD*) = 87.08 (7.76), range = 57.33–98. The Analogy Finding Task measures were both residualized with respect to the number of invalid analogies selected (see Method). The descriptive statistics for the raw Analogy Finding Task measures are as follows: Analogy Finding Task—Uncued – *M*(*SD*) = 8.98 (4.05), range = 1–16. Analogy Finding Task—Creativity Cue – *M*(*SD*) = 9.90 (4.12), range = 0–17.

Figure 1

Correlation Matrix of Ability and Academic Measures



Note. Zero-order correlation (Pearson's r) matrix of all ability measures and computer science grades. r-values that are significant at $p < .05$ are shown in black. See the online article for the color version of this figure.

Computer Science Grades ($r(130) = .224$, $p = .009$) cannot be explained by individual differences in measures of other types of reasoning, other types of creativity, or even performance on the same task in the absence of a cue to be creative.

Taken together, these results suggest that the ability to make creative analogical connections between disparate concepts is associated with grades earned in university computer science courses, and this association cannot be explained by differences in other forms of creativity (i.e., generating new ideas) or by domain-general reasoning abilities.

Do Attitudes Related to Creativity Predict Computer Science Grades?

We next assessed the extent to which attitudes related to creativity were associated with grades earned in computer science courses. In particular, we asked whether Creativity Anxiety and/or CompSci-

Table 2

Multiple Regression Model Predicting Computer Science Grades From Ability Measures

Predictor	β	SE	t	p
AUT Fluency	.006	.096	.06	.950
AUT Originality	.096	.098	.98	.329
RAT	−.037	.105	−.35	.727
Analogy Finding Task—Uncued	−.110	.131	−.84	.402
Analogy Finding Task—Creativity Cue	.262	.133	1.98	.050
Ravens	.008	.097	.83	.407
Syllogisms	.039	.093	.42	.677
Course Level	−.127	.098	−1.29	.199

Note. AUT = Alternative Uses Task; RAT = Remotes Associate Task. Model $F(8, 123) = 1.40$, $p = .203$. Overall adjusted $R^2 = .024$.

Creativity Overlap were associated with Computer Science Grades. We also included Noncreativity Anxiety Control scores, Trait Anxiety, and Computer Science Anxiety as covariates. For descriptive statistics of all attitude measures, see Table 3. Results displayed in Figure 2 show that while CompSci-Creativity Overlap was significantly associated with Computer Science Grades ($r(130) = .233, p = .007$), Creativity Anxiety was not ($r(130) = .006, p = .941$). Replicating previous work (Wilson and Shrock, 2001), Computer Science Anxiety was also shown to be associated with Computer Science Grades ($r(130) = -.210, p = .015$).

We next sought to test whether CompSci-Creativity Overlap predicted unique variance in Computer Science Grades even when all other attitude measures were controlled for. Results in Table 4 show that CompSci-Creativity Overlap continues to predict unique variance in Computer Science Grades even when controlling for attitudes toward computer science and creativity ($\beta = .233, t(125) = 2.77, p = .006$). Computer Science Anxiety also continued to predict unique variance when controlling for all other attitudes ($\beta = -.220, t(125) = -2.55, p = .012$]. Together, these results suggest that, even when accounting for other attitudes toward creativity and computer science, the more students perceive computer science as creative, the better grades they earned.

Post Hoc Analyses: Considering Ability and Attitude Measures Together

In the previous sections, we found evidence that several variables predicted success in university-level computer science courses. Here, we wished to assess whether the variance in computer science grades explained by these measures was shared or whether they explained unique variance in computer science grades. To test this, we ran a multiple regression model predicting computer science grades that included all variables that were found to predict significant variance in Computer Science Grades in the previous analyses: Analogy Finding Task—Creativity Cue, CompSci-Creativity Overlap, and Computer Science Anxiety. Because the effect of the creativity cue on analogy matrix performance was of interest, we also included Analogy Finding Task—Uncued as a covariate. To account for differences in grades that can be explained by the level of the computer science course (intro-level vs upper-level), we also included Course Level as a covariate. This model predicting computer science grades included the following variables: Analogy Finding Task—Creativity Cue, CompSci-Creativity Overlap, Computer Science Anxiety, Analogy Finding Task—Uncued, and Course Level. Results in Table 5 show that each of the measures that were found to predict computer science grades in earlier analyses predicted unique variance in computer science grades. This suggests that each of the variables identified may be independently important in shaping success in university-level computer science grades.

Moderation Analyses

Finally, we tested whether associations between creativity measures and Computer Science Grades were moderated by either Course Level or CompSci-Creativity Overlap. First considering Course Level as a moderator, we hypothesized that measures related to creativity may matter more for upper-level students compared with intro-level students. To test this hypothesis, we ran separate regression models in which the interaction between Course Level and the following creativity measures were assessed: AUT Fluency, AUT Originality, RAT, Analogy Finding Task—Creativity Cue, Creativity Anxiety, and CompSci-Creativity Overlap. Results showed that Course Level did not significantly moderate the associations between any of these measures and Computer Science Grades (all interaction term ps $> .10$), providing no evidence to suggest that measures of creativity have differential predictive effects on computer science grades for intro-level students compared with upper-level students.

Next considering CompSci-Creativity Overlap as a moderator, we hypothesized that students who perceived computer science as creative to a greater degree would show strengthened associations between measures of creativity and Computer Science Grades. To test this hypothesis, we ran separate regression models in which the interaction between CompSci-Creativity Overlap and each of the

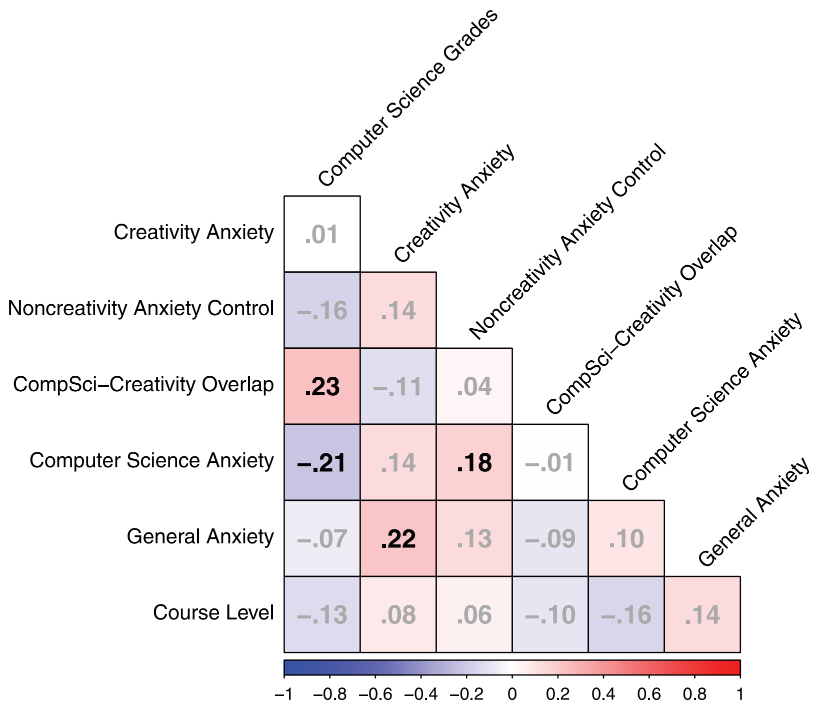
Table 3
Descriptive Statistics of Attitude Measures

Measure	*M* (*SD*)	Range
Creativity Anxiety	12.1 (6.38)	0–28
Noncreativity Anxiety Control	7.61 (5.59)	0–27
CompSci-Creativity Overlap	5.19 (1.25)	2–7
Computer Science Anxiety	5.05 (2.41)	1–10
Trait Anxiety	44.59 (8.48)	20–70

Note. Table 3 shows descriptive statistics of all attitude measures.

Figure 2

Correlation Matrix of Attitude and Academic Measures



Note. Zero-order correlation (Pearson's *r*) matrix of all ability measures and computer science grades. *r*-values that are significant at $p < .05$ are shown in black. See the online article for the color version of this figure.

creativity measures listed in the previous analysis were assessed. In each of these regression models, Course Level was entered as a covariate. Results showed that CompSci-Creativity Overlap did not significantly moderate the associations between these measures and Computer Science Grades (all interaction term *ps* > .20), providing no evidence to suggest that measures of creativity differentially predict success in computer science courses for those who perceive computer science as more creative.

Discussion

The present study sought to investigate whether creative ability measures and measures of attitudes related to creativity could predict success in introductory and advanced-level computer science courses. While computer scientists and computer science education researchers largely view computer science as a creative discipline (Knobelsdorf, & Romeike, 2008; Romeike, 2007a, 2007b), no study to our knowledge

Table 4

Multiple Regression Model Predicting Computer Science Grades From Attitude Measures

Predictor	*β*	*SE*	*t*	*p*
Creativity Anxiety	.096	.086	1.11	.269
Noncreativity Anxiety Control	−.138	.085	−1.62	.108
CompSci-Creativity Overlap	.233	.084	2.77	.006
Computer Science Anxiety	−.220	.086	−2.55	.012
Trait Anxiety	−.008	.086	−.09	.929
Course Level	−.145	.086	−1.70	.092

Note. Model $F(6, 125) = 3.50$, $p = .003$. Overall adjusted $R^2 = .103$.

Table 5
Multiple Regression Model Predicting Computer Science Grades From Ability and Attitude Measures

Predictor	β	*SE*	*t*	*p*
Analogy Finding Task—Creativity Cue	.249	.122	2.05	.042
CompSci-Creativity Overlap	.191	.083	2.31	.023
Computer Science Anxiety	−.234	.083	−2.82	.006
Analogy Finding Task—Uncued	−.078	.121	−.65	.517
Course Level	−.132	.084	−1.58	.116

Note. Model $F(5, 126) = 4.73$, $p < .001$. Overall adjusted $R^2 = .125$.

has directly examined whether creativity measures can predict achievement in computer science courses. As creativity is one of the "core soft skills" of computer science (Romeike, 2007a), identifying the specific creative factors that predict computer science success could be an important step in developing effective curricula that emphasize creativity in the computer science classroom.

Of the creative ability measures examined, performance on the analogy finding task when cued to be creative was the lone predictor of computer science grades. When all ability measures were entered into a regression model together, performance on the Analogy Finding Task when cued to be creative predicted unique variance in computer science grades over and above all other measures. The model controlled for performance on a version of the Analogy Matrix Task that *did not* prompt participants to be creative, signifying that the more participants were able to respond to the creative cue by approaching the same task in a more creative way, the better they were able to do in their computer science course. While future work should be done to understand whether this effect would replicate in samples at other universities, this finding suggests that the ability to make connections between seemingly unrelated concepts—specifically when prompted to think creatively—may help to support student success in computer science courses. The ability measures regression model also controlled for measures of more domain-general reasoning abilities such as nonverbal and verbal reasoning (as measured by Ravens and Syllogisms) that did not predict performance in computer science courses. The inclusion of these domain-general reasoning abilities rules out possible noncreative confounds, and additionally controlling for creative idea generation (AUT performance) and creative insight ability (RAT performance) suggests the resulting predictive association is specific to creative analogical reasoning.

What might explain why creative analogical reasoning, even over and above other forms of creativity, predicts computer science grades? Analogical connections are found by identifying similarities between the relational information structures of analogs (i.e., the way that elements of information within one analog relate to each other is similar to the way that elements of information within the other analog relate to each other; Gick & Holyoak, 1980; Green et al., 2006; Markman & Gentner, 2000). Making connections between information structures that are not obviously alike on the surface is important for aspects of computer science, like software design, that require the ability to creatively combine core principles to solve challenging, higher-level problems (Romeike, 2007a). Indeed, students are initially taught fundamental computer science principles and core concepts, including programming basics like recursion, principles of hierarchization such as nesting, and the structure/syntax of the given programming language (Romeike, 2007a; Schwill, 1997). Students utilize these core concepts as "building blocks," or a set of well understood concepts, which can be combined to produce complex programs. The challenge of computer science lies in this combination, as success comes from one's ability to make connections between the demands of an assignment and the building blocks that can be creatively combined to meet those demands. Doing this well requires both the ability to find connections between what students know (building blocks) and the constraints of the problem (which blocks can be used), and how to creatively combine these blocks to solve the problem.

On exam questions in computer science courses, students are often presented with a problem that is superficially different from problems they have encountered before, but that in fact can be solved by using concepts they learned about in a different context in class.

For a concrete example from an exam question used in the upper-level course included in this study—Intro to Network Security—students are asked to come up with a solution to fix a vulnerability in a specific type of encryption used in wireless communication (WiFi). The question is open-ended, but the most straightforward solution involves noticing that a concept they learned about in a completely different context in the course—a means to authenticate that messages came from a specific source—could also be used in this new context. In this example, students had to identify that the relationship between the different elements of the problem presented were the same in important ways as relationships between other concepts they had learned in class. In other words, they had to engage in creative analogical reasoning to make the connection between the current problem and previous solutions, even when the superficial elements differed substantially between the current situation and the previously encountered one. Crucially then, creative analogical reasoning skills, or the ability to identify connections between seemingly distant concepts, might aid students in solving these types of particularly difficult problems. Another reason why creative analogical reasoning ability might predict computer science achievement while other types of creativity (measured by the AUT and RAT) do not is that creative analogical reasoning contains substantial elements of both divergent and convergent thinking, relative to other measures that emphasize either divergence (e.g., AUT) or convergence (e.g., RAT) more strongly (Cortes et al., 2019; Green et al., 2010). Speculatively, this combination of divergence with convergence might better reflect the kind creativity that contributes to computer science performance. Future work could further explore this idea by examining whether performance on different creativity tasks predicts performance on individual exam items that vary with respect to their demands on convergent or divergent reasoning. Additional research that builds on the present findings could also be done to explore whether other types of creative abilities may be predictive of computer science success in addition to creative analogical reasoning. Given that past work has shown that many successful programmers use visuospatial strategies when planning their code (Adelson, 1981; Cooke & Schvaneveldt, 1988; McKeithen et al., 1981), it is possible that measures of creativity within this domain (e.g., the Torrance test of Creative Thinking—Figural; Torrance, 1974) may also be associated with computer science achievement, especially measures that tap into the right combination of divergent and convergent processes.

We next assessed the extent to which attitudes related to creativity predicted grades in computer science courses. We found no evidence to suggest that individual differences in creativity anxiety predicted computer science grades. This suggests that attitudes toward creativity itself may have little bearing on success in the types of computer science courses studied here. However, we found that perceptions of the overlap of creativity and computer science predicted grades earned in computer science courses, where students who perceived a greater degree of overlap between computer science and creativity performed better in computer science courses. Additionally, when all attitude measures were entered into the same regression model, perceptions of the overlap of creativity and computer science continued to predict unique variance in computer science grades. Why might this be the case? One intriguing possibility is that students who perceive computer science as creative find computer science more interesting and invest more time into the discipline, improving their understanding and ability. Indeed, students tend to prefer working on aspects of computer science that they view as more creative (Gu & Tong, 2004), and including opportunities for students to be creative in computer science classrooms is closely related to motivation and understanding of the material (Romeike, 2006). Therefore, students who perceive computer science as involving creativity might spend more time engaging with the subject creatively, increasing their success. Importantly then, computer science educators who highlight the creative nature of multiple aspects of computer science and allow for creative exploration of these topics may raise interest and subsequently increase success in their classrooms.

Another mechanism that might explain this finding is that students who perceive a large overlap between creativity and computer science may approach the subject in a more creative way. These students may attempt to find creative connections and identify creative solutions to problems rather than simply trying to strictly apply what they have been taught in the past. Taking a creative approach might precipitate more success when solving difficult computer science problems (Scragg et al., 1994). Though the present findings are entirely correlational and cannot provide definitive insights into mechanisms underlying the observed correlations, they nonetheless suggest that, if college-level instructors and university departments are able to encourage the perception that computer science is creative, they may improve the performance of their students. Short of running an intervention-based study, future work could address the question of whether those who perceive computer science as creative are more likely to approach computer science in a creative way by having computer science students complete individual problems while measuring the strategies they use to complete those problems. If this interpretation is correct, then students who indicate perceiving a higher degree of overlap between computer science and creativity would be more likely to use creative problem-solving strategies when approaching computer science problems. This proposed future work could also aim to better understand what factors (for instance, personality factors like openness to experience or cognitive factors like intelligence or even various forms of creative abilities) are associated with a tendency to use creative problem-solving strategies or to be more likely to see various activities as creative.

In addition to our hypothesis that perceptions of the overlap of creativity and computer science might predict performance in computer science courses, we predicted that these perceptions might moderate the influence of creative factors on performance. Contrary to our hypothesis, we found no evidence that perceptions of involvement of creativity in computer science moderated any relations between our other creativity measures and computer science performance. Additionally, we hypothesized that measures of creative ability, anxiety about being creative, and perceived involvement of creativity in computer science might differ in their level of impact on computer science success at different levels of computer science training. Again, contrary to our hypothesis, we found no evidence that course level moderated any relations between creativity measures and computer science performance. Together, these results provide no evidence to suggest that the observed associations between creativity measures and success in computer science courses are dependent on individual differences in the degree to which computer science is perceived as creative or on the level of the course itself.

An important limitation of the present work is that while we were able to collect real-world data on exam performance in computer science courses, we were not able to obtain data on performance on individual exam questions within those exams. It is almost certainly the case that different types of exam questions would differ in the extent to which creative abilities or approaches would benefit performance, with some exam questions leaving more room for creativity than others. Here, we find evidence that creative analogical reasoning ability and the extent to which individuals perceive computer science as overlapping with creativity predict overall exam scores, and it is quite possible that these associations are driven by associations with performance on specific types of exam questions. Future work should aim to examine performance on individual computer science exam questions to better understand whether measures of creativity are more strongly associated with certain types of problems than others. Doing so could provide another way to test the hypothesis generated by the present findings that those who perceive a greater overlap between computer science and creativity approach computer science problems in more creative ways—expert raters with computer science training could rate how creative responses to different exam questions are, similarly to the way that measures of originality are obtained for tasks like the Alternative Uses Task (Guilford, 1967; described in the Method section). If the hypothesis is correct, then those who report seeing a greater overlap between computer science and creativity should give more creative responses. We believe the present work provides the impetus to obtain this more fine-grained exam information in future studies on the association between creativity and computer science achievement.

It is important to note that this study was primarily exploratory in nature. Future work that tests the extent to which these findings would hold in different samples at different universities would need to be done to assess the replicability of the present findings. Furthermore, students were not randomly assigned to enroll in computer science courses and our sample was fairly self-selecting. Our findings can best be thought of as pertinent to the population of students who choose to participate in computer science classes rather than all university students. Therefore, just because individual differences in a given measure did not predict computer science grades in the present sample does not necessarily suggest that it is not important for computer science. Nevertheless, this work suggests, for the first time, that among students enrolled in computer science courses, measures of creativity can be a predictor of success. This research may contribute to improving the efficacy of computer science courses. Evidence that performance in computer science courses is predicted by creative analogical reasoning ability, and by perceptions of computer science as creative, points to the potential efficacy of creativity-related interventions in research and classroom contexts. In particular, past meta-analytic work has shown that creative abilities can be effectively trained and that effects of creativity training can be long-lasting (Scott et al., 2004). If it is the case that creative analogical thinking can similarly be trained, then such training may transfer to increased computer science success. Additionally, researchers and educators interested in boosting success in computer science courses can test whether emphasizing the creative nature of computer science can be an effective means of improving student outcomes. Finally, we believe that this work could be used to inform perceptions of computer science itself. As we noted in in the beginning of the article, many computer scientists themselves consider computer science to be a creative field (Knobelsdorf & Romeike, 2008; Romeike, 2007a), and much of what happens in computer science writ large (even outside the classroom) involves the generation of novel and useful content, placing these activities squarely within the definition of creativity that most creativity researchers use (Runco & Jaeger, 2012). Our current findings showing that specific measures of creativity are associated with computer science achievement, at least within the confines of a university-level computer science course, provides additional empirical support that computer science should, indeed, be thought of as a creative field.

References

Aasheim, C., Shropshire, J., Li, L., & Kadlec, C. (2019). Knowledge and skill requirements for entry-level IT workers: A longitudinal study. *Journal of Information Systems Education*, *23*(2), 8.

Adelson, B. (1981). Problem solving and the development of abstract categories in programming languages. *Memory & cognition*, *9*(4), 422–433.

Amabile, T. M. (1988). A model of creativity and innovation in organizations. *Research in Organizational Behavior*, *10*(1), 123–167.

Apiola, M., Lattu, M., & Pasanen, T. A. (2010). Creativity and intrinsic motivation in computer science education: Experimenting with robots. In *Proceedings of the fifteenth annual conference on innovation and technology in computer science education* (pp. 199–203). Association for Computing Machinery. https://doi.org/10.1145/1822090.1822147

Aron, A., Aron, E. N., & Smollan, D. (1992). Inclusion of other in the self scale and the structure of interpersonal closeness. *Journal of Personality and Social Psychology*, *63*(4), 596–612. https://doi.org/10.1037/0022-3514.63.4.596

Baer, M. (2012). Putting creativity to work: The implementation of creative ideas in organizations. *Academy of Management Journal*, *55*(5), 1102–1119. https://doi.org/10.5465/amj.2009.0470

Beaty, R. E., & Johnson, D. R. (2020). Automating creativity assessment with *SemDis*: An open platform for computing semantic distance. *Behavior Research Methods*, *53*, 757–780.

Bergin, S., & Reilly, R. (2005). Programming: Factors that influence success. In *Proceedings of the 36th SIGCSE technical symposium on computer science education* (pp. 411–415). ACM Press.

Bowden, E. M., & Jung-Beeman, M. (2003). Normative data for 144 compound remote associate problems. *Behavior Research Methods, Instruments, & Computers*, *35*(4), 634–639. https://doi.org/10.3758/BF03195543

Bureau of Labor Statistics, U.S. Bureau of Labor Statistics. (n.d.). *Occupational outlook handbook*. Computer and Information Research Scientists. https://www.bls.gov/ooh/computer-and-information-technology/computer-and-information-research-scientists.htm

Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. In *Proceedings of the 6th annual conference on innovation and technology in computer science education* (pp. 49–52). https://doi.org/10.1145/377435.377467

Cennamo, K., Brandt, C., Scott, B., Douglas, S., McGrath, M., Reimer, Y., & Vernon, M. (2011). Managing the complexity of design problems through studio-based learning. *The Interdisciplinary Journal of Problem-Based Learning*, *5*(2), 5. https://doi.org/10.7771/1541-5015.1253

Cennamo, K., Douglas, S. A., Vernon, M., Brandt, C., Scott, B., Reimer, Y., & McGrath, M. (2011). Promoting creativity in the computer science design studio. In *Proceedings of the 42nd ACM technical symposium on computer science education* (pp. 649–654). https://doi.org/10.1145/1953163.1953344

Charlton, J. P., & Birkett, P. E. (1999). An integrative model of factors related to computing course performance. *Journal of Educational Computing Research*, *20*(3), 237–257. https://doi.org/10.2190/BTG0-7VQK-6XD3-G4C4

Cortes, R. A., Weinberger, A. B., Daker, R. J., & Green, A. E. (2019). Re-examining prominent measures of divergent and convergent creativity. *Current Opinion in Behavioral Sciences*, *27*, 90–93. https://doi.org/10.1016/j.cobeha.2018.09.017

Cooke, N. J., & Schvaneveldt, R. W. (1988). Effects of computer programming experience on network representations of abstract programming concepts. *International Journal of Man-Machine Studies*, *29*(4), 407–427.

Cortes, R. A., Weinberger, A. B., Colaizzi, G. A., Porter, G., Dyke, E., Keaton, H., Walker, D. L., & Green, A. E. (2021). What makes mental modeling difficult? Normative data for the Multidmensional Relational Reasoning Task (MRRT). *Frontiers in Psychology*, *12*, 1512.

Csikszentmihalyi, M., & Sawyer, K. (2014). Creative insight: The social dimension of a solitary moment. In R. J. Sternberg & J. E. Davidson (Eds.), *The systems* *model of creativity* (pp. 73–98). Springer. https://doi.org/10.1007/978-94-017-9085-7_7

Daker, R. J., Cortes, R. A., Lyons, I. M., & Green, A. E. (2020). Creativity anxiety: Evidence for anxiety that is specific to creative thinking, from STEM to the arts. *Journal of Experimental Psychology: General*, *149*(1), 42–57. https://doi.org/10.1037/xge0000630

Evans, G. E., & Simkin, M. G. (1989). What best predicts computer proficiency? *Communications of the ACM*, *32*(11), 1322–1327. https://doi.org/10.1145/68814.68817

Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology*, *12*(3), 306–355. https://doi.org/10.1016/0010-0285(80)90013-4

Glass, R. L. (2006). *Software Creativity 2.0*. developer.* Books.

Green, A. E. (2016). Creativity, within reason: Semantic distance and dynamic state creativity in relational thinking and reasoning. *Current Directions in Psychological Science*, *25*(1), 28–35. https://doi.org/10.1177/0963721415618485

Green, A. E., Fugelsang, J. A., & Dunbar, K. N. (2006). Automatic activation of categorical and abstract analogical relations in analogical reasoning. *Memory & Cognition*, *34*(7), 1414–1421. https://doi.org/10.3758/BF03195906

Green, A. E., Kraemer, D. J., Fugelsang, J. A., Gray, J. R., & Dunbar, K. N. (2010). Connecting long distance: Semantic distance in analogical reasoning modulates frontopolar cortex activity. *Cerebral Cortex*, *20*(1), 70–76. https://doi.org/10.1093/cercor/bhp081

Green, A. E., Spiegel, K. A., Giangrande, E. J., Weinberger, A. B., Gallagher, N. M., & Turkeltaub, P. E. (2017). Thinking cap plus thinking zap: tDCS of frontopolar cortex improves creative analogical reasoning and facilitates conscious augmentation of state creativity in verb generation. *Cerebral Cortex*, *27*(4), 2628–2639. https://doi.org/10.1093/cercor/bhw080

Gu, M., & Tong, X. (2004). Towards hypotheses on creativity in software development. In F. Bomarius & H. Iida (Eds.), *International conference on product focused software process improvement* (pp. 47–61). Springer.

Guilford, J. P. (1967). *The nature of human intelligence*. McGraw-Hill.

Hill, A. M. (1998). Problem solving in real-life contexts: An alternative for design in technology education. *International Journal of Technology and Design Education*, *8*(3), 203–220. https://doi.org/10.1023/A:1008854926028

Knobelsdorf, M., & Romeike, R. (2008). Creativity as a pathway to computer science. In *Proceedings of the 13th annual conference on innovation and technology in computer science education* (pp. 286–290). Association for Computing Machinery. https://doi.org/10.1145/1384271.1384347

Konvalina, J., Wileman, S. A., & Stephens, L. J. (1983). Math proficiency: A key to success for computer science students. *Communications of the ACM*, *26*(5), 377–382. https://doi.org/10.1145/69586.358140

Kounios, J., & Beeman, M. (2015). *The Eureka factor: Creative insights and the brain*. Random House.

Kurtz, B. L. (1980, February). Investigating the relationship between the development of abstract reasoning and performance in an introductory programming class. In *Proceedings of the eleventh SIGCSE technical symposium on Computer science education* (pp. 110–117).

Leach, R. J., & Ayers, C. A. (2005). *The psychology of invention in computer science*. PPIG.

Leeper, R. R., & Silver, J. L. (1982). Predicting success in a first programming course. *ACM SIGCSE Bulletin*, *14*(1), 147–150. https://doi.org/10.1145/953051.801357

Markman, A. B., & Gentner, D. (2000). Structure mapping in the comparison process. *The American Journal of Psychology*, *113*(4), 501–538. https://doi.org/10.2307/1423470

McKeithen, K. B., Reitman, J. S., Rueter, H. H., & Hirtle, S. C. (1981). Knowledge organization and skill differences in computer programmers. *Cognitive Psychology*, *13*(3), 307–325.

Mednick, S. A. (1962). The associative basis of the creative process. *Psychological Review*, *69*(3), 220–232. https://doi.org/10.1037/h0048850

Necka, E. A., Sokolowski, H. M., & Lyons, I. M. (2015). The role of self-math overlap in understanding math anxiety and the relation between math anxiety and performance. *Frontiers in Psychology*, *6*, 1543. https://doi.org/10.3389/fpsyg.2015.01543

Núñez-Peña, M. I., Guilera, G., & Suárez-Pellicioni, M. (2014). The single-item math anxiety scale: An alternative way of measuring mathematical anxiety. *Journal of Psychoeducational Assessment*, *32*(4), 306–317. https://doi.org/10.1177/0734282913508528

Peteranetz, M. S., Flanigan, A. E., Shell, D. F., & Soh, L. K. (2017). Computational creativity exercises: An avenue for promoting learning in computer science. *IEEE Transactions on Education*, *60*(4), 305–313. https://doi.org/10.1109/TE.2017.2705152

Peteranetz, M. S., Flanigan, A. E., Shell, D. F., & Soh, L. K. (2018). Helping engineering students learn in introductory computer science (CS1) using computational creativity exercises (CCEs). *IEEE Transactions on Education*, *61*(3), 195–203. https://doi.org/10.1109/TE.2018.2804350

Peteranetz, M. S., Wang, S., Shell, D. F., Flanigan, A. E., & Soh, L. K. (2018). Examining the impact of computational creativity exercises on college computer science students' learning, achievement, self-efficacy, and creativity. In *Proceedings of the 49th ACM technical symposium on computer science education* (pp. 155–160). Association for Computing Machinery. https://doi.org/10.1145/3159450.3159459

Petrone, P. (2019). *The skills companies need most in 2019 – And how to learn them*. Linked In. https://www.linkedin.com/pulse/skills-companies-need-most-2019-how-learn-them-paul-petrone

Pfeiffer, W. (2019, June 18). *There are 700,00 open tech jobs. Here is how firms are hiring for them*. CNBC. https://www.cnbc.com/2019/06/18/there-are-70000-open-tech-jobs-here-is-how-firms-are-hiring-for-them.html

Prabhakaran, R., Green, A. E., & Gray, J. R. (2014). Thin slices of creativity: Using single-word utterances to assess creative cognition. *Behavior Research Methods*, *46*(3), 641–659. https://doi.org/10.3758/s13428-013-0401-7

Raven, J., Raven, J. R., & Court, J. H. (1998). *Raven's progressive matrices and vocabulary scales*. Oxford Psychologists Press.

Ren, Z., Daker, R. J., Shi, L., Sun, J., Beaty, R. E., Wu, X., Chen, Q., Yang, W., Lyons, I. M., Green, A. E., & Qiu, J. (2021). Connectome-based predictive modeling of creativity anxiety. *NeuroImage*, *225*, 117469. https://doi.org/10.1016/j.neuroimage.2020.117469

Romeike, R. (2006). Creative students: What can we learn from them for teaching computer science? In *Proceedings of the 6th Baltic Sea conference on computing education research: Koli Calling 2006* (pp. 149–150). Association for Computing Machinery.

Romeike, R. (2007a). Three drivers for creativity in computer science education. In *Proceedings of the IFIP Conference on "Informatics, mathematics and ICT: A golden triangle"*. Boston.

Romeike, R. (2007b). Applying creativity in CS high school education: criteria, teaching example and evaluation. In *Proceedings of the Seventh Baltic Sea conference on computing education research-Volume 88* (pp. 87–96). Australian Computer Society, Inc.

Runco, M. A., & Jaeger, G. J. (2012). The standard definition of creativity. *Creativity Research Journal*, *24*(1), 92–96.

Saunders, D., & Thagard, P. (2005). Creativity in computer science. In J. C. Kaufman & J. Baer (Eds.), *Creativity across domains: Faces of the muse* (pp. 153–167). Lawrence Erlbaum Associates.

Schwill, A. (1997). Fundamental ideas: Rethinking computer science education. *Learning and Leading With Technology*, *25*(1), 28–31.

Scott, G., Leritz, L. E., & Mumford, M. D. (2004). The effectiveness of creativity training: A quantitative review. *Creativity Research Journal*, *16*(4), 361–388. https://doi.org/10.1080/10400410409534549

Scragg, G., Baldwin, D., & Koomen, H. (1994). Computer science needs an insight-based curriculum. In *Proceedings of the twenty-fifth SIGCSE symposium on computer science education* (pp. 150–154). Association for Computing Machinery. https://doi.org/10.1145/191029.191092

Spielberger, C. D., Gorsuch, R. L., & Lushene, R. E. (1970). *The State-Trait Anxiety Inventory (test manual)*. Consulting Psychologists Press.

Torrance, E. P. (1974). *Torrance Test of Creative Thinking: Norms-technical manual*. Scholastic Testing Service.

Weinberger, A. B., Iyer, H., & Green, A. E. (2016). Conscious augmentation of creative state enhances "real" creativity in open-ended analogical reasoning. *PLoS ONE*, *11*(3), e0150773. https://doi.org/10.1371/journal.pone.0150773

Werth, L. H. (1986). Predicting student performance in a beginning computer science class. *ACM SIGCSE Bulletin*, *18*(1), 138–143. https://doi.org/10.1145/953055.5701

Wiedenbeck, S., Labelle, D., & Kain, V. N. (2004). *Factors affecting course outcomes in introductory programming*. PPIG.

Wiedenbeck, S. (2005). Factors affecting the success of non-majors in learning to program. In *Proceedings of the first international workshop on Computing education research* (pp. 13–24). Association for Computing Machinery.

Wilson, B. C., & Shrock, S. (2001). Contributing to success in an introductory computer science course: a study of twelve factors. *Acm Sigcse Bulletin*, *33*(1), 184–188. https://doi.org/10.1145/366413.364581

Yang, C., Huang, Q., Li, Z., Liu, K., & Hu, F. (2017). Big Data and cloud computing: Innovation opportunities and challenges. *International Journal of Digital Earth*, *10*(1), 13–53. https://doi.org/10.1080/17538947.2016.1239771

Appendix

Remote Associates Test Stimuli from Bowden and Jung-Beeman (2003)

Show/Life/Row—BOAT
Duck/Fold/Dollar—BILL
Rocking/Wheel/High—CHAIR
Loser/Throat/Spot—SORE
Preserve/Range/Tropical—FOREST
Aid/Rubber/Wagon—BAND
Flake/Mobile/Cone—SNOW
Safety/Cushion/Point—PIN
Fish/Mine/Rush—GOLD
Political/Surprise/Line—PARTY
River/Note/Account—BANK
Opera/Hand/Dish—SOAP
Print/Berry/Bird—BLUE
Pie/Luck/Belly—POT
Date/Alley/Fold—BLIND

(*Appendix continues*)

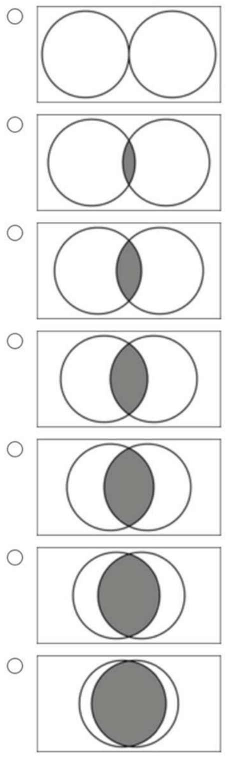
Syllogistic Reasoning Task Stimuli adapted from Cortes et al. (2021)

Condition	Dimensions	True or false	Premise 1	Premise 2	Conclusion
Spatial	1	FALSE	Mason is above and to the right of Henry.	Edward is below and neither right nor left of Mason.	Henry is to the right of Edward.
Nonspatial	2	TRUE	Caleb is more excited and neither more nor less certain than Travis.	Travis is more certain and more excited than Logan.	Logan is less certain and less excited than Caleb.
Spatial	1	FALSE	Brian is above and to the right of Caleb.	Caleb is above and to the left of Ethan.	Brian is below Ethan.
Spatial	2	TRUE	Mason is above and to the left of Peter.	Peter is above and to the left of Derek.	Derek is below and to the right of Mason.
Nonspatial	1	TRUE	Logan is less certain and more excited than Edward.	Travis is more certain and more excited than Edward.	Logan is less certain than Travis.
Spatial	2	FALSE	Mason is right of and neither above nor below Victor.	Peter is below and to the left of Victor.	Mason is below and to the left of Peter.
Nonspatial	1	TRUE	Logan is more certain and more excited than Lucas.	Logan is more certain and neither more nor less excited than Peter.	Lucas is less excited than Peter.
Nonspatial	2	FALSE	Derek is more certain and less excited than Henry.	Brian is less certain and more excited than Henry.	Brian is more certain and less excited than Derek.
Nonspatial	2	FALSE	Peter is more excited and neither more nor less certain than Logan.	Lucas is less certain and less excited than Logan.	Peter is less certain and less excited than Lucas.
Nonspatial	1	TRUE	Mason is less certain and less excited than Peter.	Derek is less certain and neither more nor less excited than Peter.	Derek is more excited than Mason.
Spatial	1	FALSE	Ethan is below and to the left of Lucas.	Lucas is above and neither right nor left of Henry.	Henry is to the left of Ethan.
Spatial	2	TRUE	Travis is below and to the left of Edward.	Edward is left of and neither above nor below Logan.	Travis is below and to the left of Logan.
Spatial	2	FALSE	Brian is below and to the right of Caleb.	Brian is above and to the left of Victor.	Caleb is below and to the left of Victor.
Nonspatial	2	TRUE	Roger is less certain and more excited than James.	James is less certain and more excited than William.	Roger is less certain and more excited than William.
Nonspatial	1	TRUE	Henry is less certain and less excited than Derek.	Henry is more certain and less excited than Brian.	Brian is less certain than Derek.

(*Appendix continues*)

Perceived Computer Science—Creativity Overlap Measure adapted from Aron, Aron, and Smollan (1992) and Necka, Sokolowski, and Lyons (2015)

How much does **Creativity** overlap with **Computer Science**:



(*Appendix continues*)

Analogy Finding Task Matrices from Weinberger et al. (2016)

Matrix 1

	Orbit Sun	Polygraph Honesty	Spark Fire	Absolution Sinner	Yardstick Distance	Planet Soil	Sweeper Street	Wax Candle	Orange Peel	Puppy Dog	Tea Honey	Cotton Sweatshirt	Dollar Fortune	Address House	Shampoo Hair	Clock Time	Cigarette Butt	Seed Tree	Scout Talent	Passport Border
Watermelon Rind	□	□	□	□	□	■	□	□	■	□	□	□	□	□	□	□	■	□	□	□
Soap Hand	□	□	□	■	□	□	■	□	□	□	□	□	□	□	■	□	□	□	□	□
Wool Sweater	□	□	□	□	□	□	□	■	□	□	□	■	■	□	□	□	□	□	□	□
Ruler Length	□	■	□	□	■	□	□	□	□	□	□	□	□	□	□	■	□	□	■	□
Kitten Cat	□	□	■	□	□	□	□	□	□	■	□	□	■	□	□	□	□	■	□	□

Matrix 2

	Mitt Baseball	Amnesia Memory	Boat Anchor	Inventor Invention	Eraser Pencil	Spray Skunk	Mother Daughter	Antenna Signal	Kneepad Knee	Porkchop Pig	State Country	Sleep Pajamas	Chapter Book	Atmosphere Earth	Battery Toy	Cloud Rain	Water Flame	Tongue Taste	Decade Lifetime	Road Map
Helmet Head	■	□	□	□	□	■	□	□	■	□	□	□	□	■	□	□	□	□	□	□
Whiteout Pen	□	■	□	□	■	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□
Nose Scent	■	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	■	□	□
Father Son	□	□	□	■	□	□	■	□	□	□	□	□	□	□	□	■	□	□	□	□
Lambchop Lamb	□	□	□	□	□	□	□	□	□	■	■	□	■	□	□	□	□	□	■	□

Received December 8, 2020
Revision received June 2, 2021
Accepted June 3, 2021 ■