Using Rasch Analysis for Determining the Cut Score of a Computer Science Placement Exam

Steven McGee, The Learning Partnership
Everett Smith, EVS Psychometric Services
Andrew Rasmussen, Jeremy Gubman, Chicago Public Schools

Abstract

A key strategy for broadening computer science participation in the Chicago Public Schools has been the enactment of a yearlong computer science course as a high school graduation requirement. The *Exploring Computer Science* (ECS) curriculum and professional development program serves as a core foundation for supporting policy enactment. However, students with prior background in computer science might find the course repetitive. This paper reports on district efforts to develop a placement exam for students to take an advanced computer science course in lieu of the introductory computer science course. The placement exam tasks were modeled after the ECS exam tasks but with higher difficulty. We used Rasch modeling to equate the placement exam tasks to the ECS exams and to establish a cut score for passing the placement exam.

1 Introduction

A key strategy for broadening computer science participation in the Chicago Public Schools (CPS), where a majority of students are Latinx or African American, has been the enactment of a high school computer science graduation requirement in 2016 (Dettori, et al., 2018). *Exploring Computer Science* (ECS) is the primary course that students have been taking to fulfill the graduation requirement in CPS. The ECS curriculum is composed of activities that engage students in computer science inquiry around meaningful projects (Margolis et al., 2012). The pedagogy of ECS is structured around three interwoven strands: equity, inquiry, and computer science concepts. The accompanying ECS professional development program and instructional coaching prepares teachers to implement these inquiry-based activities. It emphasizes building a classroom culture that is culturally responsive and adapting lessons to the backgrounds and interests of the students (Goode, Margolis, & Chapman, 2014).

CPS began implementation of ECS in 2012 (Margolis, et al., 2013), which laid the foundation for the enactment of the computer science graduation requirement. Since 2012, 65,682 students have completed the ECS course. Three-quarters of those students doing so since the enactment of the graduation policy. The demographics of ECS students have been reflective of overall demographics of CPS high school students, with 42% African American and 42% Latinx (McGee, McGee-Tekula, Dettori et al., 2019). In spring 2020, roughly 14,000 students fulfilled the graduation requirement by completing one-year of computer science.

The Chicago Alliance For Equity in Computer Science (CAFÉCS) was formalized as an RPP in 2017 to ensure that CPS provides schools with sufficient support and accountability so that the *quality* of computer science course experiences is equitable across the district. CAFÉCS is a research-practice partnership among the Chicago Public Schools (CPS), DePaul University, Loyola University, The Learning Partnership, and the University of Illinois Chicago. To ensure that all students in Chicago participate in engaging, relevant, and rigorous computing experiences, CAFÉCS addresses problems of practice through research and development that increases opportunities for all students to pursue computing pathways and prepares all students for the future of work. CAFÉCS has contributed to the literature on the positive impact of the ECS course on students' attitudes towards computer science (McGee, et al., 2017; McGee, McGee-Tekula, Duck, McGee, et al., 2018) as well as students' choices about future computer science coursework (McGee et. al., 2017, McGee, McGee-Tekula, Duck, Dettori,

et al., 2018). In addition, the results in Chicago have shown that students are achieving equivalent posttest performance of computational thinking in ECS across all races, ethnicities, and genders (McGee, McGee-Tekula, Duck, McGee, et al., 2018).

In addition to the work in supporting high school computer science, CPS has been expanding support for computer science in the elementary schools in Chicago. The rapid growth of computer science in both high schools and elementary schools has spawned new challenges. In particular, an increasing number of students were entering high school with prior experience in computer science, for whom ECS would seem repetitive. The district leaders determined that a placement exam is one piece of data that schools could use to identify students who were prepared to take an advanced computer science course as the means to fulfill the graduation requirement.

2 Assessment Framework

The development of the placement exam builds upon prior work by Snow, Tate, Rutstein, and Bienkowski (2017) on the development of course exams for the ECS curriculum. These course exams were developed using the Evidence-Centered Design framework, which is an assessment methodology that is especially advantageous when the knowledge and skills to be measured involve complex, multistep performances (Mislevy & Haertel, 2006). The Evidence-Centered Design is a three-step process. (1) Working with various stakeholders, the assessment developers identified the important computational thinking practices in the ECS curriculum. This first step in the process resulted in an assessment framework that articulated the specific computational practices from ECS that would be the focus of the assessment. These practices were distributed across the four core units within ECS (Human-Computer Interaction, Problem Solving, Web Development, and Programming). (2) The assessment developers mapped the computational thinking practices in the framework to a model of evidence that can support inferences about those practices. The model specified the kinds of assessment tasks and the features of those tasks that would serve as evidence. (3) Lastly, the developers created specific tasks that would elicit that evidence. This process results in a set of tasks which comprised a pretest, end of unit assessments and a posttest for the ECS course. The assessments were field tested and validated with 941 students over two years (See Snow, Rutstein, Bienkowski, & Xu, 2017 for details). Based on the results of the field test, CPS adopted the ECS pretest and posttest assessments and teachers routinely administer them as part of their implementation of ECS (McGee, McGee-Tekula, Duck, McGee, et al., 2018). The results have shown that posttest performance is equivalent across races, ethnicities, and genders.

As discussed above, the district leaders decided to develop a placement exam as one piece of evidence that schools could use to identify students who were prepared to take an advanced computer science course as the means to fulfill the graduation requirement. Given the positive benefit for students who completed the ECS course, it was decided that students should only place out of ECS if they had a strong background in computer science. We decided that the level of performance on the placement exam to be considered for placing out of the introductory course should be equivalent to getting an A in the ECS class.

In summer 2017, we gathered a team of three ECS teachers and two subject matter experts from the CPS Office of Science. Two members of the SRI assessment development team (Rutstein and Tate) came to Chicago and facilitated a three-day item writing training workshop. At the workshop, teachers were introduced to the assessment framework and examined assessment questions from the ECS exams that aligned to the components of the framework. The SRI team also provided a general introduction to item writing. Teachers were then assigned particular parts of the assessment framework to develop items for. Each teacher developed one assessment task that was reviewed by the team and by the SRI facilitators. The team provided feedback to the teachers in terms of the alignment to the framework, the quality of the task, the meaningfulness of the task for students, and the level of difficulty.

During the remaining part of the summer, the teachers independently developed the assessment questions, which were each reviewed by the subject matter experts in the CPS Office of Computer Science. While the initial item writing was guided by the SRI assessment team, the CPS Office of Computer Science subject matter experts and researchers at The Learning Partnership led the bulk of the

development and validation of the placement exam task. Therefore, the development of the district's computer science placement exam serves as a validation of the viability of the use of the assessment framework by assessment developers who were not involved in the development of the framework.

There was a total of 12 tasks that were successfully reviewed and field tested. In order to validate the new assessment tasks with the ECS assessment tasks, it was necessary to include tasks from the ECS assessments on the placement exam forms. Given the level of effort to complete each task and the limitation of administering the field test forms within one class period, we decided to have no more than six tasks on one form. We created four field test forms with each placement exam task appearing on one of the four forms. We selected tasks from the ECS exams to appear on the various forms to serve as linking items to be able to link the four forms to each other and to the ECS assessments. Table 1 provides details on which items appeared on each form. The posttest contained one task in common with the pretest. The placement exam pool contained 4 tasks in common with the pretest and one task in common with the posttest.

In spring 2018, the field test forms were completed by 257 students who were enrolled in advanced computer science courses in CPS. We hired The Graide Network for scoring the items on the field test forms. The Graide Network is the same organization that managed the scoring process for the ECS pretest and posttest assessments. The Graide Network recruited and trained three undergraduate preservice teachers to score the performances tasks. They were provided training on each of the rubrics prior to scoring. As part of the training, each scorer was expected to accurately score a benchmark set of assessments to establish their understanding of the rubrics. In order to link the scorers and minimize the number of times each task was scored, we had overlapping subsets of scorers rate the same students. We used the Facets software version 3.71.4 to conduct Many-Facet Rasch Measurement analysis (MFRM) to scale the student responses at each administration. Facets develops a model based on how well the student performed across the range of question prompts with set difficulties taking into account the severity of the scorer relative to the other scorers. Within MFRM, the goal is not for scorers to arrive at agreement on the scores, but instead to model the variation in how the scorers interpreted the rubrics. As long as the raters are internally consistent in how they apply the rubric, Facets can adjust the students' scores based on the severity or leniency of the scores relative to other scorers.

3 Methods

There are three sets of data corresponding to the ECS pretest, ECS posttest, and the placement exam forms. We wanted to use the responses across the three sets of data to place all of the items on a common metric for future use (linking data via common items is a frequently used methodology within educational testing for the development of item banks, for example, in computer adaptive testing or the creation of parallel forms – see Wright (1977) and Smith and Stone (2009)). The first two sets of data were from an implementation study of ECS involving 4640 students who completed the ECS pretest and 2769 students who completed the ECS posttest. The third set of data came from the field test of the placement exam items using 4 forms (n=257).

Given these three sets of data, our goal was to combine all the data on the metric of the ECS posttest data. We chose the ECS posttest data for the metric to link all other data as the decisions to be made regarding future course placements would be based on the assumption that students are being assessed on the content taught in the course. Given this, our sequence of analyses is to 1) analyze the ECS post data to establish the underlying metric, 2) to anchor (fix) the model estimates on the post data results and analyze the ECS pretest data (the anchoring process forces the ECS pretest data on the same metric as the ECS posttest data), and finally, 3) to anchor on the results of combined posttest and pretest ECS data and analyze the district placement data.

To accomplish these analyses, we needed to use a methodology that would not only enable us to link assessments containing common items, but also a methodology that would take into account the severity of raters (the degree to which a given rater assigns scores that are on average higher or lower than other raters) employed for data collection where not every rater encounters every student. The Many-Facets Rasch model (MFRM) is well suited for these purposes (Linacre, 1989). The MFRM does not

require that every student be rated by every rater on every item. It is only necessary that data collection be conducted so that every parameter can be linked to every other parameter by some connecting observations (Linacre, 1989). This enables all model parameters to be placed on one common metric.

The MFRM is an extension of the Rasch dichotomous model (Rasch, 1980; Wright and Masters, 1982; Wright and Stone, 1979) that can be used for rater-mediated assessments. The program used for the Rasch analysis is Facets (Linacre, 2014), which uses PROX and JMLE estimation methods. For the estimation of model parameters, the ratings given by raters are the source of data. For example, the sum of the ratings awarded to a particular student across all raters and items is the sufficient statistic for estimation of student proficiency. Likewise, the sum of ratings by a given rater across all students and items is the sufficient statistic for estimation of rater severity.

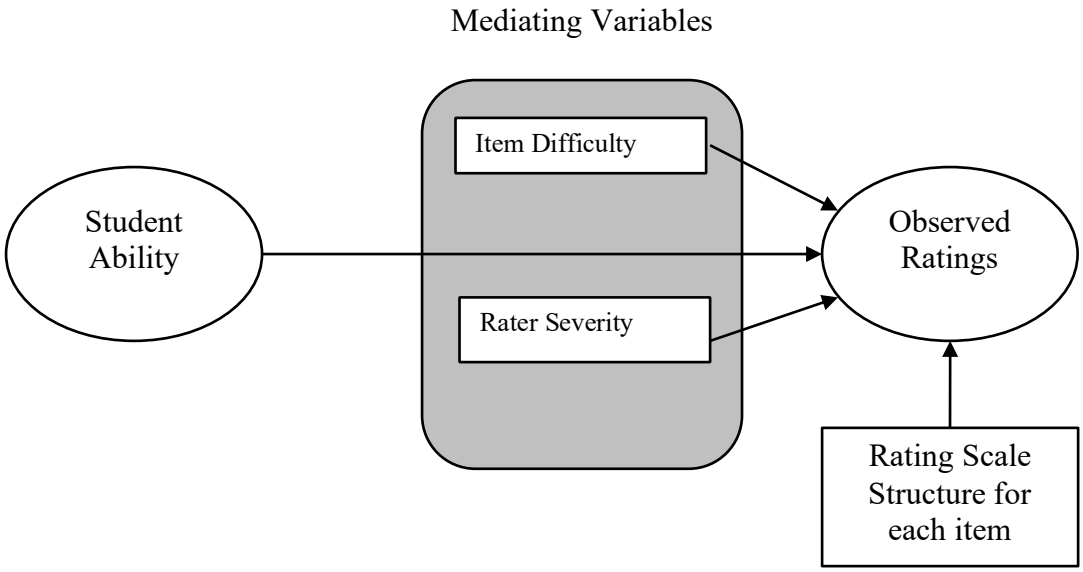
The specific model used for the current analysis is a three-facet Rasch partial credit model (students, raters, and items). The partial credit model was required rather than the more parsimonious rating scale model as each item has its own scoring criteria (see Smith & Smith, 2004). This model is given by equation (1) and graphically represented in Figure 1.

$$\ln [(P_{nijk}) / (P_{nijk-1}) = B_n – T_i – C_j - F_{ik} \quad (1)$$

where

P_{nijk} is the probability of student n on item i being rating k by rater j
P_{nijk-1} is the probability of student n on item i being rating k-1 by rater j
B_n is the proficiency level of student n
T_i is the difficulty of item i
C_j is the severity of rater j
F_{ik} is the threshold of being rated in category k versus k -1 on item i

Figure 1
Graphically representation of the model used



The unit of measurement resulting from the Facets analysis is the logit ('log odd units'). When the data fits the model assumptions, these logits are on an interval scale, making the estimates appropriate for further parametric statistical analysis (Merbitz, Morris, and Grip, 1989; Wright, 1977; Wright and Linacre, 1989). Greater logit values for items and raters indicate increasing item difficulty and rater severity, respectively. Students with higher logit values demonstrate higher levels of performance than students at lower levels. There are two model assumptions. First is unidimensionality, which implies that all of the assessment tasks are measuring only one underlying construct. Therefore, increasing amounts of the underlying construct can be represented on a linear scale. Second is local independence, which means that after controlling for the underlying traits, ratings are statistically independent of each other. These assumptions can be evaluated using the fit statistics available with the Facets program.

Once the parameters of the MFRM are estimated, they are used to compute expected responses. Various fit statistics (item, rater, student) are then derived from a comparison of the expected and observed responses. Facets provides two types of fit statistics: Infit and Outfit. Our focus will be on Outfit as it has been shown to be statistically more powerful to detect more types of measurement disturbances than Infit (Smith & Smith, 2004). When reported as mean-square statistics (MNSQ), Outfit values are simply chi-square statistics divided by the degrees of freedom. The expected value of the outfit metric is 1, but it can range from 0 to ∞ (Wright and Linacre, 1994). Outfit values below the expected value of 1 suggest a violation of the local independence since the data lack stochasticity or randomness. Outfit values greater than 1 represent unmodeled excessive variability, which may imply that more than one dimension is being measured, which would signify a departure from assumption of unidimensionality.

Criteria for evaluating adequate fit depends on the consequence of test score use and the type of data being collected with rater/judge-mediated data typically having less stringent criteria due to the more subjective nature of rater data compared to, for example, data obtained under high-stakes standardized testing conditions. Values ranging from 0.6 to 1.4 are indicative of good fit (Wright and Linacre, 1994)

for self-report data; however, Smith (2005) demonstrated excessive variability is more a threat to productive measurement in terms of impacting the measurement of students. Given the exploratory nature of our investigation and the comparatively low consequences to test score use, we will note items or raters with Outfit mean-square fit statistics greater than 1.5. Large misfit for items may indicate that the item was scored in an inconsistent manner by one or more raters for one or more students. This may be due to, for example, an ambiguous scoring criterion. In a similar manner, large misfit for raters may indicate that the rater was scoring in an inconsistent manner across one or more students for one or more items which may be due to, for example, inconsistent use of the scoring criteria due to insufficient training.

To provide an example of how a possible cut score could be established using this bank of items to predict the likely future success in advanced computer science courses using actual course grades and the relationship of these course grades to the item bank difficulties, we anchor on the final results from combining all three data sets and then add to the model course grades. In our demonstration, we use two recodes of course grades to reflect "success" in the course: grade of A versus all other grades and grades of A or B versus all other grades. Given the common metric of student ability and the fact that we treat course grade as an "item" (successful versus not successful), we demonstrate how the interaction of student ability and the difficulty of the item representing course grade can provide a probability of achieving an A in the course. Depending on what probability is used, different levels of student ability may be viable cut scores to implement in future applications of this assessment to place students in the advanced computer science courses.

4 Results

Figure 2 provides a visual representation of the distributions of student ability, item difficulty, and rater severity on the posttest. The first column is the logit metric. Student contains the estimates of the student abilities, with higher values indicating more ability. Higher logit values for items and raters indicate more difficult items and more severe raters. Visually we see the students are spread out in terms of their ability estimates on the ECS posttest (Person reliability of .76). The item labeled Post2 is the easiest item to receive high ratings on. The item labeled Pre1 is the hardest. The rater ID shows that raters do differ in their severity levels (chi-square of 653.3 (16), $p < .01$) and hence statistical methods to take into account this variability are warranted. With regard to the MNSQ Outfit statistics, all values for items were less than 1.30 and all values for raters less than 1.35.

Anchoring on the values obtained from the calibration of the ECS Posttest data, Figure 3 shows the location of the additional tasks from the ECS Pretest on the ECS Posttest metric. A visual check of the items from Figure 2 shows their locations in Figure 3 remain unchanged as a result of the anchoring process. Students are still spread out in terms of their ability estimates. The student-level reliability estimate increased slightly to .77. The item labeled Pre3 is now the easiest item to receive high ratings on and the item labeled Pre1 still the hardest. The overall distribution of pretest and posttest item difficulty levels is similar, which is to be expected since the forms were created to be equivalent. The rater ID again shows raters do differ in their severity levels (chi-square 4671.2 (36), $p < .01$). One item (Pre3) and two raters had MNSQ Outfit statistics above the 1.5 threshold.

Continuing with our goal of a larger item pool, we anchored on the values obtained from the calibration of the ECS pretest/posttest data and added the placement exam data from the four field test forms. Figure 4 shows the location of the additional items from the placement exam data on the ECS posttest metric. A visual check of the items from Figure 3 shows the locations of the ECS pretest and ECS posttest items in Figure 4 remain unchanged as a result of the anchoring process. Students are still spread out in terms of their ability estimates. The student-level reliability remained at .77. The item labeled Pre3 remains the easiest item to receive high ratings on with several of the new placement items now being the most difficult. Most of the placement exam tasks are at least as difficult or more difficult than the most difficult pretest and posttest tasks, which is to be expected since the placement exam questions were written to be more difficult. Four items (Pre3, Place1, Place2, and Place5) and two raters had MNSQ Outfit statistics above the 1.5 threshold.

As an exploration of how to relate the items to observed course grades, our final calibration was to add in two indicators of success in the CPS ECS courses. We created two dichotomous indicators based on observed course grades (gradeA = A in course versus all other grades and gradeAB = A or B in course versus all other grades). The location of these two indicators is highlighted in Figure 5. The difficulty of gradeA is .50 logits. The difficulty for gradeAB is -1.14 logits. This ordering makes sense as it shows it is easier to get an A or B (gradeAB) than it is to get an A (gradeA).

To model possible cuts scores for future applications of the assessment we utilize the dichotomous Rasch model given in equation (2), where the probability of passing the course is a function of student's ability (β) and gradeA (or gradeAB; δ) difficulty:

$$\{X_{ni} = Pass\} = \frac{e^{\beta n - \delta_i}}{1 + e^{\beta n - \delta_i}}, \quad (2)$$

Given δ is fixed at .50 (gradeA) or -1.14 (gradeAB), a simple spreadsheet can determine the probabilities of passing for various levels of student ability. Table 2 provides a truncated version of the list of full outcomes to show some examples. The first column contains various levels of student ability based on the final pool of items. Columns two and four contain the difficulty levels of getting an A (gradeA difficulty) or getting an A or B (gradeAB difficulty). Application of equation (2) yields the probabilities for success of getting an A (column 3) or getting an A or B (column 5). For example, if a committee determines that based on the results of a placement test, a student must have a 0.80 probability of achieving an A the introductory ECS course, the student would need to show a proficiency of 1.89 logits (see highlighted items in Table 2). Similarly, for a student to have a 0.90 probability of achieving an A in the ECS course, the student would need to score 2.70 logits on the placement exam.

5 Conclusion

The Chicago Public Schools has achieved significant success in broadening participation in computing through the enactment of a high school computer science graduation requirement. Most students have fulfilled the requirement through the introductory ECS course, which has been shown to support equivalent student outcomes by race, ethnicity and gender. In addition, ECS inspires students to take additional high school computer science coursework. CPS has also increased the integration of computer science across elementary schools in the district. Therefore, there is an increasing number of students entering high school with significant amounts of computer science experience. For some of those students, the introductory ECS course would be redundant. In this paper, we described the process of developing and validating a pool of assessment tasks that could inform the development of a placement exam. The placement exam would assess the extent to which the students demonstrate competency in computational thinking practices that is equivalent to the level that students achieved who received an A in the class.

With the initial support of the developers of the ECS pretest and posttest, we developed twelve assessment tasks that cover the range of topics in the ECS curriculum. These tasks were placed on four field test forms with four ECS pretest and posttest tasks serving as linking items. The placement exam tasks were scored by an independent organization in which overlapping subsets of three scorers rated the same students. This same scoring process was previously used to score the ECS pretests and posttests. A total of 33 scorers were used across the pretest, posttest, and placement exam field test. Two of these raters exhibited misfit slightly above the target number. We can use features found in Facets to determine which ratings are the likely sources of the misfit. For example, a rater may have unexpectedly given a very high rating to a difficult item to a student with lower overall ability. Identifying the source of the misfit may help with future rater training. For example, we may find the misfit being due to some possible ambiguity in the rating directions for a particular item.

For the most part, the collection of 21 assessment tasks (9 pretest/posttest tasks and 12 placement exam tasks) had Infit and Outfit statistics within an acceptable range indicating that with the exception of

three placement exam items with fit statistics slightly above the target, the remaining placement exam items met the assumptions of unidimensionality and local independence.

These analyses have been used to create a placement exam with 8 tasks. The goal of the selection process was to create a placement exam that had two questions for each chapter, at least one linking item from the ECS pretest and/or posttest and have difficulty levels that are close to potential cutscores reflecting the performance level consistent with getting an A in the ECS class. Using the 0.80 probability as a potential cutscore, there were a number of potential placement exam items that were within 1 logit of the logit value corresponding to 0.80 probability of achieving an A (1.89). These items would provide higher measurement precision. We also included items that were much easier items when they rounded out the necessary content coverage. Even though they would have lower precision for determining the cutscore, they signal to teachers and students that the content is important.

In our future work, we will field test and validate the placement exam as an intact form, linked to ECS pretest and posttest performance in order to establish a cutscore. In preliminary discussions with schools that would be using the placement exam, we have found that the metric of achieving an A in the class has been confusing to explain, as some stakeholders have thought the score meant that it is predicting what the students would get if they took ECS. Whereas the statistic is simply comparing their current level of knowledge to someone who completed the class and received an A. In our future work, we will be comparing the placement exam performance to predict future performance in advanced computer science classes. Rather than provide a specific cutscore, we will provide an overall probability passing the computer science AP exam.

6 Acknowledgements

The authors were supported in part by the National Science Foundation grants CNS-1738572 to The Learning Partnership and CNS-1738776 to DePaul University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

Suggested Citation

McGee, S., Smith, E., Rasmussen, A. M., and Gubman, J. (2021, April 8-12). *Using Rasch analysis for determining the cut score of a computer science placement exam* [Paper presentation]. American Educational Research Association, online. https://doi.org/10.51420/conf.2021.4

References

Brennan, R. L. (Ed.). (2006). *Educational Measurement*. Rowman & Littlefield.

Dettori, L., Greenberg, R. I., McGee, S., Reed, D., Wilkerson, B., and Yanek, D. (2018) CS as a graduation requirement: Catalyst for systemic change. In *Proceedings of the 49th SIGCSE Technical Symposium on Computer Science Education*, pages 406–407. Association for Computing Machinery, Baltimore, MD. https://doi.org/10.1145/3159450.3159646.

Linacre, J.M. (1989). *Multi-facet Rasch measurement.* Chicago: MESA Press

Linacre, J.M. (2014). *FACETS Rasch measurement computer program.* Chicago: Winsteps.com.

Liu, M. and Haertel, G. (2011). Design Patterns: A Tool to Support Assessment Task Authoring. Large-Scale Assessment Technical Report 11, Menlo Park, CA: SRI International.

Merbitz, C., Morris, J., and Grip, J.C. (1989). Ordinal scales and foundations of misinference. *Archives of Physical Medicine and Rehabilitation, 70,* 308-312.

Mislevy, R. J., Hamel, L., Fried, R., Gaffney, T., Haertel, G, Hafter, A.,…Wenk, A. (2003). *Design patterns for assessing science inquiry* (PADI Technical Report 1). Menlo Park, CA: SRI International.

Mislevy, R. J. and Haertel, G.D. (2006). Implications of evidence-centered design for educational testing. *Educational Measurement: Issues and Practice 25*(4), 6–20.

McGee, S., McGee-Tekula, R., Dettori, L., Rasmussen, A. M., & Greenberg, R. I. (2019). Research Methods for Reaching Urban Students from Groups Underrepresented in STEM Disciplines. In T. Ruecker & V. Svihla (Eds.), Research, Interrupted: Confronting and Overcoming Challenges in Education Research. London: Routledge.

McGee, S., McGee-Tekula, R., Duck, J., Dettori, L., Greenberg, R.I., Reed, D.F., Wilkerson, B., Yanek, D., & Rasmussen, A.M. (2018, April). Does Exploring Computer Science Increase Computer Science Enrollment? Paper presented at the American Education Research Association annual meeting, New York.

McGee, S., McGee-Tekula, R., Duck, J., McGee, C., Dettori, L., Greenberg, R.I., Snow, E., Rutstein, D., Reed, D., Wilkerson, B., Yanek, D., Rasmussen, A., & Brylow, D. (2018, February). Equal Outcomes 4 All: A study of student learning in Exploring Computer Science. Proceedings of SIGCSE '18, Baltimore, MD, USA, 50-55. doi: 10.1145/3159450.3159529

McGee, S., McGee-Tekula, R., Duck, J., White, T., Greenberg, R. I., Dettori, L., Reed, D. F., Wilkerson, B., Yanek, D., Rasmussen, A.M., & Chapman, G. (2017). Does a Taste of Computing Increase Computer Science Enrollment? Computing in Science & Engineering, 19(3), 8-18.

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests.* Copenhagen: Danmarks Pædagogiske Institut. (Chicago: University of Chicago Press, 1980).

Smith, Jr., E. V. (2001). Evidence for the reliability of measures and the validity of measure interpretation: A Rasch measurement perspective. *Journal of Applied Measurement, 2*, 281-331.

Smith, Jr., E.V. (2005). Effect of item redundancy on Rasch item and person estimates. *Journal of Applied Measurement, 6*, 147-163.

Smith, Jr., E.V., and Smith, R.M. (2004). *Introduction to Rasch measurement.* Maple Grove, MN: JAM Press.

Smith, Jr., E.V., and Smith, R.M. (2007). *Rasch measurement: Advanced and specialized applications.* Maple Grove, MN: JAM Press.

Smith, Jr., E.V., and Stone, G.E. (2009). *Criterion-Referenced Testing: Practice Analysis to Score Reporting using Rasch Measurement Models.* Maple Grove, MN: JAM Press.

Snow, E., Rutstein, D., Bienkowski,M., and Xu, Y. (2017). Principled Assessment of Student Learning in High School Computer Science. In ICER '17 Proceedings of the 2017 ACM Conference on International Computing Education Research. 209–216.

Snow, E., Tate, C., Rutstein, D., and Bienkowski, M. (2017). Assessment Design Patterns for Computational Thinking Practices in Exploring Computer Science. Technical Report. SRI International.

Wolfe, E.W., & Smith, Jr. E.V. (2007a). Instrument development tools and activities for measure validation using Rasch models: Part I—Instrument Development Tools. *Journal of Applied Measurement, 8,* 97-123.

Wolfe, E.W., & Smith, Jr. E.V. (2007b). Instrument development tools and activities for measure validation using Rasch models: Part II—Validation Activities. *Journal of Applied Measurement, 8,* 204-233.

Wright, B.D. (1977). Solving measurement problems with the Rasch model. *Journal of Educational Measurement, 14,* 97-116.

Wright, B.D., and Linacre, J.M. (1989). Observations are always ordinal; Measurements, however, must be interval. *Archives of Physical Medicine and Rehabilitation, 70,* 857-860.

Wright, B.D, and Linacre, M. (1994). Reasonable mean-square fit values. *Rasch Measurement Transactions, 8,* 370.

Wright, B,D., and Masters, G.N. (1982). *Rating scale analysis: Rasch measurement.* Chicago: MESA Press.

Wright, B.D., and Stone, M.H. (1979). *Best Test Design.* Chicago: MESA Press.

Table 1: Distribution of tasks on forms showing the interconnectedness of the assessment forms

Tasks	ECS Pretest	ECS Posttest	Form 1	Form 2	Form 3	Form 4
Pre1	•	•		•	•	
Pre2	•					
Pre3	•					
Pre4	•					•
Pre5	•	•	•	•	•	
Pre6	•		•			
Post2		•				
Post3		•				
Post4		•	•	•	•	•
Place1			•			
Place2						•
Place3			•			
Place4				•		
Place5					•	
Place6				•		
Place7					•	
Place8						•
Place9			•	•		
Place11					•	
Place12						•
Place13						•

Table 2. Examples of probabilities of passing for gradeA and gradeAB.

Student ability	gradeA difficulty	Probability of passing	gradeAB difficulty	Probability of passing
0.20	0.50	0.43	-1.14	0.79
0.21	0.50	0.43	-1.14	0.79
0.22	0.50	0.43	-1.14	0.80
0.24	0.50	0.44	-1.14	0.80
0.25	0.50	0.44	-1.14	0.80
0.27	0.50	0.44	-1.14	0.80
0.28	0.50	0.45	-1.14	0.81
1.02	0.50	0.63	-1.14	0.90
1.04	0.50	0.63	-1.14	0.90
1.05	0.50	0.63	-1.14	0.90
1.06	0.50	0.64	-1.14	0.90
1.08	0.50	0.64	-1.14	0.90
1.88	0.50	0.80	-1.14	0.95
1.89	0.50	0.80	-1.14	0.95
1.90	0.50	0.80	-1.14	0.95
1.92	0.50	0.81	-1.14	0.96
1.93	0.50	0.81	-1.14	0.96
2.21	0.50	0.85	-1.14	0.97
2.67	0.50	0.90	-1.14	0.98
2.69	0.50	0.90	-1.14	0.98
2.70	0.50	0.90	-1.14	0.98
2.72	0.50	0.90	-1.14	0.98
2.73	0.50	0.90	-1.14	0.98

Figure 2. ECS POST results

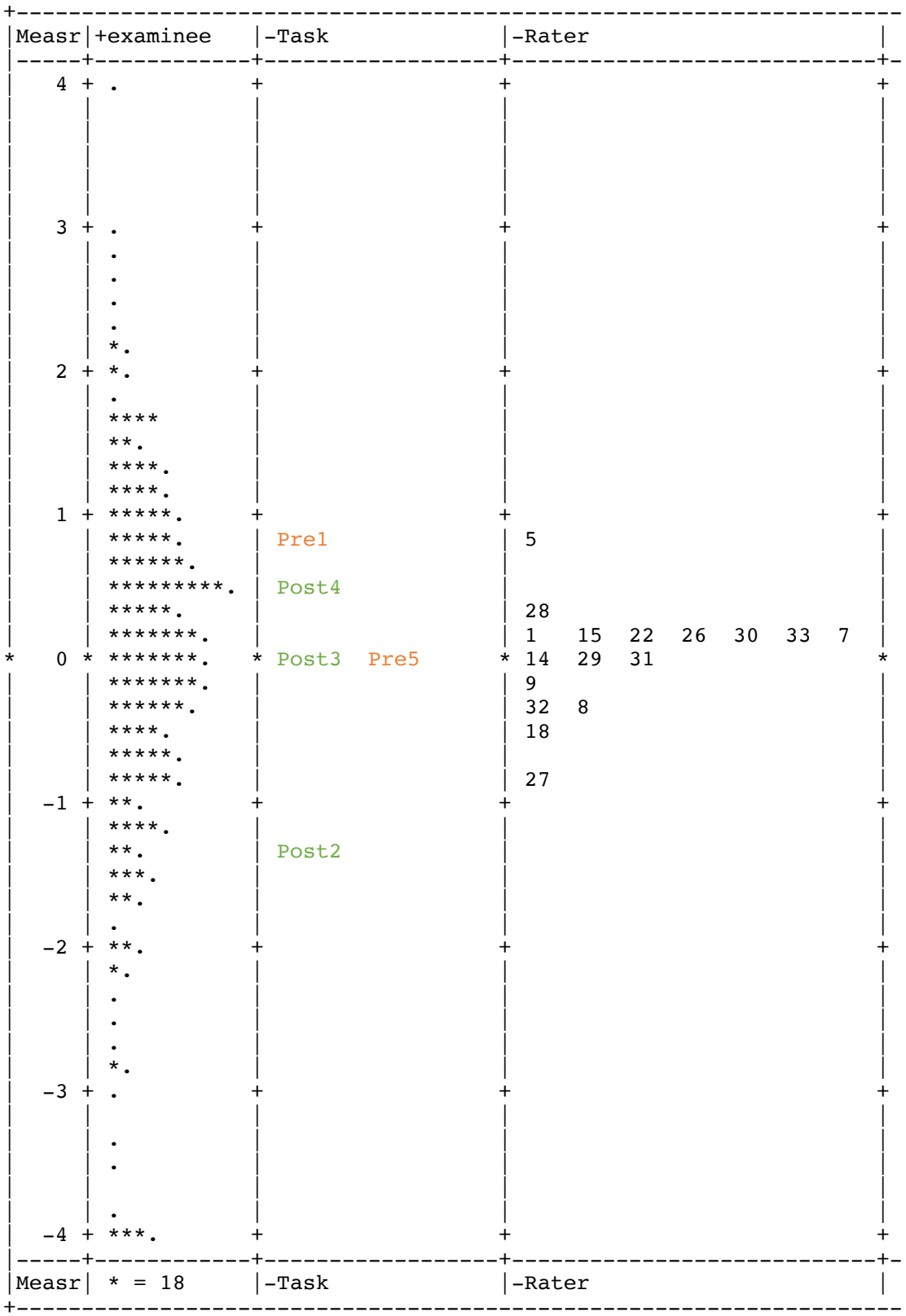


Figure 3. ECS PRE results anchored on ECE Post values

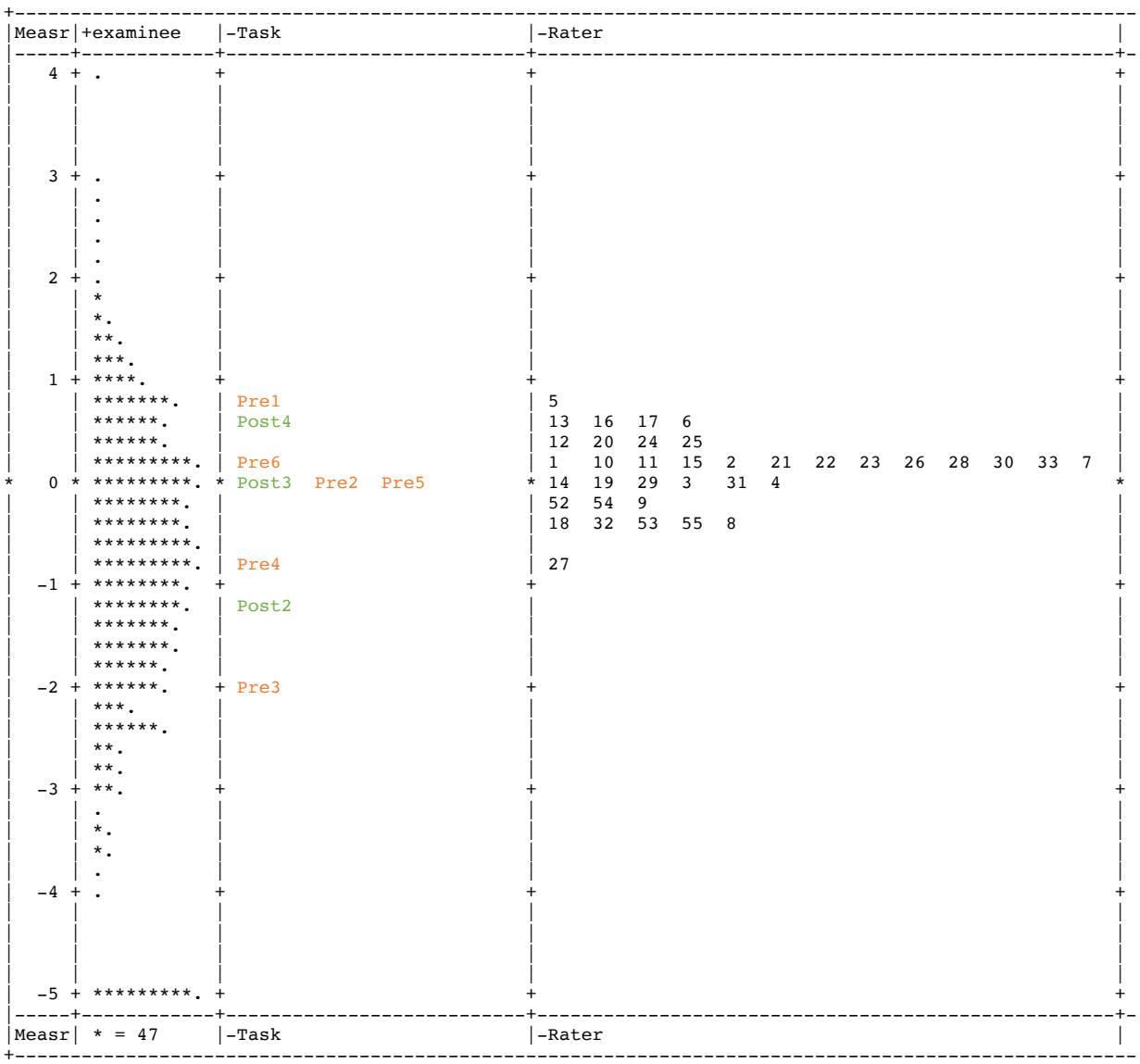


Figure 4. PLACEMENT and ECS PRE results anchored on ECS Post metric

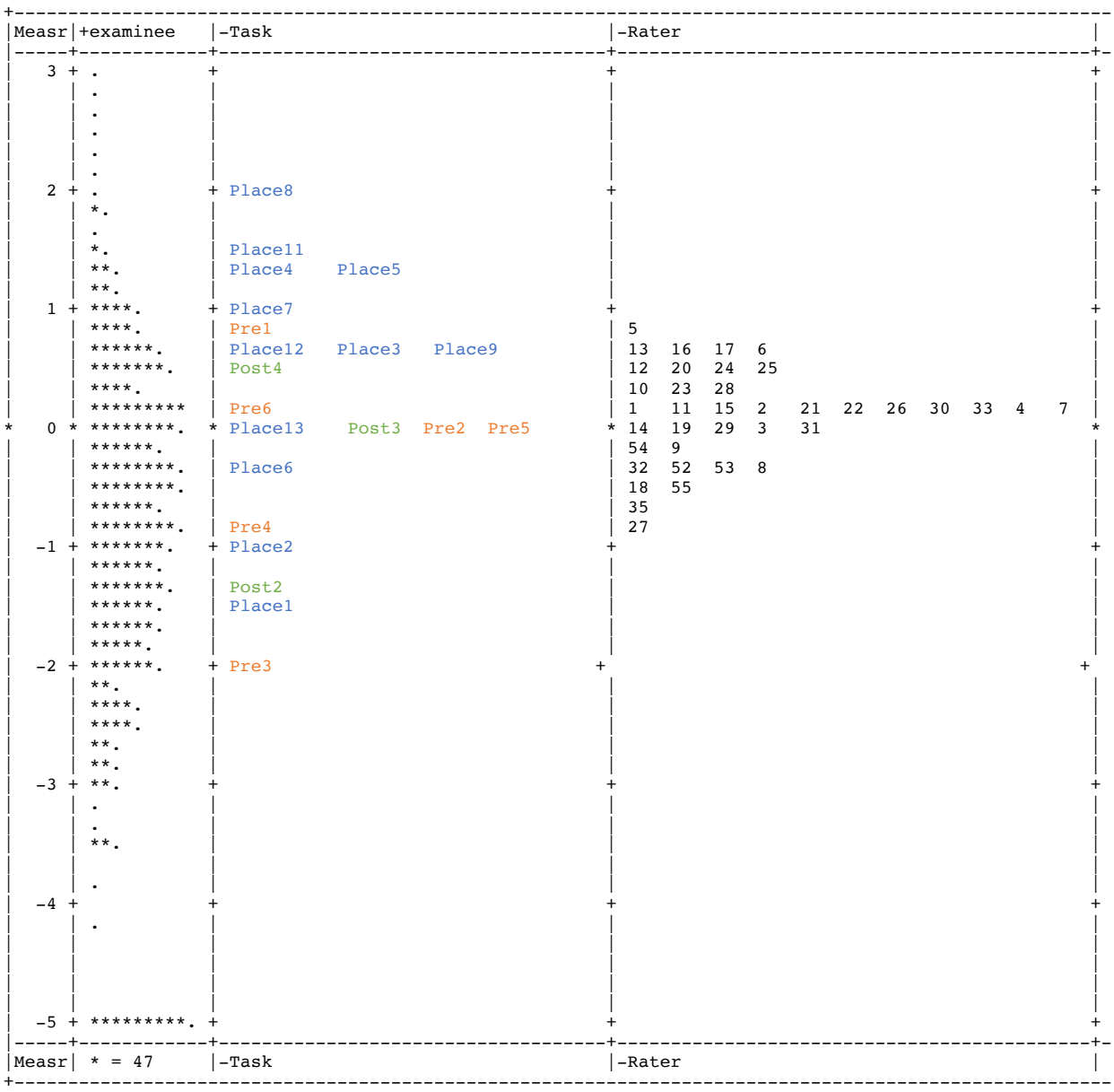


Figure 5: Course grades mapped onto the item pool metric