scoping review [16]. Both authors coded all publications; we did not calculate inter-rater reliability.

- Analysis step 1, derived initial codes based on the publications' abstracts;
- Analysis step 2, each author coded all publications and created new codes as they arose; each author re-coded previous publications with all new codes
- Analysis step 3, discussed coding and resolved any disagreements, usually adding rather than removing codes; and
- Analysis step 4, generated themes and meta-themes

3 Publications' Characteristics

To provide a high-level overview, we describe some characteristics of the publications. As shown in Figure 2, the 54 publications were published between 2006 and 2020, with the majority between 2018-2020. The publications are authored by people who work in 15 countries/regions – with the most publications from the USA (28), China (11), and India (7) – and nine of the 54 are co-authored by international collaborators. Thirty-four of the publications focus on undergraduate education while 14 pertain to K-12 education. The remainder focused on adults, all ages, or had unspecified students/participants. Thirty-six publications describe research projects, works-in-progress, or proposals. Fifteen papers are experience reports. Twenty-five of the research publications collected quantitative data and 22 collected qualitative data.

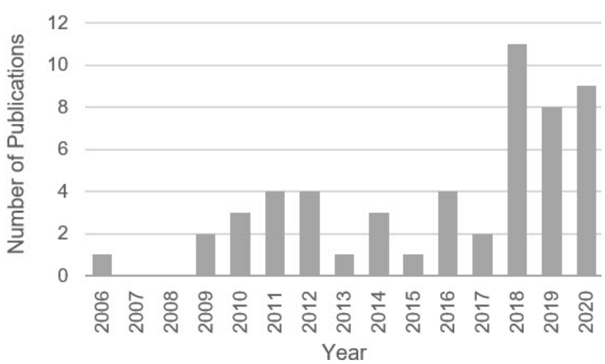

Figure 2: Number of publications by year

4 Publications' Themes

During analysis, we identified 11 themes that we grouped into three meta-themes: studying what benefits or hinders ELL students, focusing on integrative language skills, and pedagogical and curricular approaches. Ten of the themes fit into the meta-themes, while the theme of tools stands alone.

4.1 Studying What Benefits or Hinders ELL Students

4.1.1 Comparing Language of Instruction Nine publications examined the effect of the language of instruction on student learning or student experience [31, 37–40, 49–52]. All but one [31] were conducted in India.

In his PhD thesis, Pal describes a series of five projects investigating teaching programming to vernacular medium students [37], namely those students who studied in their local language at their K-12 school and in English at university. These projects were conducted in India with first-year undergraduate students whose native language is Hindi. We provide further detail about the three projects that are included in this scoping review. In these studies, Pal and Iyer compared post-test scores after a learning experience that was conducted in Hindi or English [38–40]. In each study, they used three groups: HH (Hindi in high school, Hindi in the experiment); HE (Hindi/English); and EE (English/English). In the first study, they investigated post-test scores after students learned from 1-hour video lectures [38]. They did not find any significant differences. In the second study, they investigated the impact of screencasts' medium of instruction (Hindi or English) on student performance and found a statistically significant difference in post-test scores with the HH group outperforming the HE group [40]. In the third study, they taught introductory programming workshops via English or Hindi in a classroom or via screencasts. In both the classroom and screencast contexts, post-test scores were worse for the HE group, and further, the self-paced screencast group did better than the classroom group [39].

Soosai Raj and collaborators [49–52] have conducted a series of experiments in India for native Tamil speakers investigating the effect of instructional design that combines students' native language and English when learning programming, topics in data structures, and topics in operating systems. They measured student learning via pre-test and post-tests scores and found no difference between the group taught with Tamil and English in comparison with the group taught in English. For the Tamil and English group, they used code-switching [21] – switching between English and Tamil– and translanguaging – "the process by which bilingual students and teachers engage in complex discursive practices in order to "make sense" of, and communicate in, multilingual classrooms" [23, p. 299]. They found that most students who are taught in Tamil and English will ask questions in Tamil. They conducted a sentiment analysis of student feedback and found that students had positive sentiments about the bilingual teaching methods [49]. They also found that students who studied in Tamil and English expressed positive emotions more strongly that the students who studied in English alone [49]. One important difference in comparison to Pal et al.'s work was that Soosai et al.'s student population had almost all studied at an English high school.

Lau and Yuen studied the effect of language of instruction in nine secondary schools in Hong Kong [31]. They found that students who learned in Chinese, their native language, tended to outperform students who learned in English, and that this difference was exacerbated for middle- and low-achieving students.

4.1.2 Correlates with Success in Computing Five publications focus on factors that correlate with success in computing [2, 3, 10, 42, 53].

Rauchas et al. investigated factors that correlate with success in computing courses and found that past English language performance was the best predictor of success in computer science [42]. Their work is situated in South Africa with first-year computer science students and their findings are relevant to the entire population of computer science students, not just ELL students. Ben Idris and Ammar surveyed IT and computing students and their lecturers at two Libyan universities and found that both groups believe students' English abilities affect their programming performance with English programming languages [10].

Veerasamy and Shillabeer proposed to investigate the relationship between English Language test scores and programming ability [53]. Aldmour and Nylén proposed to study whether Saudi students' English language skills transfer to an understanding of computing concepts [3]. Alaofi proposed to investigate the impact of non-native English speakers' English abilities on their programming performance [2].

4.1.3 Barriers and Difficulties Six publications explored the theme of barriers or difficulties that English Language Learners faced when learning computer science [1, 4, 18, 25, 26, 44].

Two studies were conducted with online data that was not related to a particular course at a particular institution [25, 44]. Guo surveyed users of the Python Tutor (pythontutor.com) to learn what barriers English Language Learners face when using English-language instructional materials and how they hope the materials will be improved [25]. He found that the survey respondents reported barriers with all modes of communication (reading code and documentation, writing code, listening, and speaking) and would prefer that English-language based resources used simpler language and avoided examples from any particular culture.

Reestman and Dorn investigated the Java compilation data logs from the BlueJ Blackbox database to determine whether the compiler errors generated by non-native English speakers differed from those generated by English speakers [44]. While they did find a statistically significant difference in the distribution of errors, the effect size was weak.

Hartshorn et al. surveyed instructors from five disciplines to investigate their perceptions of ELL students' language skills, reading requirements in their majors and the importance of reading, and how prepared ELL students are for their studies and future work [26]. They found that instructors perceive language skills to be more important in business and psychology than in computer science. The top three reading challenges for ELL were the same for all five majors – 'English is their second language', 'vocabulary', and 'understanding disciplinary content'. Across all majors, instructors believed that their ELL students were equally or slightly less prepared for graduate school and the workforce.

Al Zumor describes results from a survey of computing, medical, and engineering students at a Saudi Arabian university about English-medium instruction (EMI). He found that students perceive EMI to have negative academic and emotional impacts [1].

Alharbi conducted a qualitative study of international ELL students to learn what difficulties they face when studying computer science and found a number of barriers, such as understanding lectures, participating verbally, reading, and writing [4].

Feijóo-García et al. describe a study that compares native English speakers to native Spanish speakers who are learning English when using an English or Spanish Scratch interface [18]. Hispanic participants said they would prefer to use a Spanish interface.

4.2 Focusing on Integrative Language Skills

4.2.1 Bilingual Teaching in China Nine experience reports are written by authors who work in China and focus on bilingual teaching [15, 20, 30, 32, 33, 59–62]. Since 2001, the Chinese Ministry of Education has required that universities and colleges offer 5-10% of their courses bilingually [20]. The experience reports generally describe the course context, the textbook, the division of time spent teaching in English and Chinese, and what the authors learned from the experience.

The students in all courses described are native Chinese speakers. The courses generally choose English textbooks written by professors who work at world-renowned institutions. They state that far fewer textbooks are available in China than outside China, which leaves professors with limited choices that are difficult to match with the intended curriculum [62]. The textbooks are described by many authors as a challenge as they are difficult for the students to read and understand [15, 20, 30, 32, 59, 62]. Multiple authors suggest creating custom course materials to supplement the textbook [15, 20, 59], and Wang reported that the students comment positively about custom course materials [59].

Authors commonly report that teaching in English is challenging, as faculty need strong English language skills and a solid foundation in the course's technical content [20, 30, 32, 60–62]. It is not explicitly reported, but our assumption is that the professors' native language is Chinese. Feng, Xiong et al., and Jiang et al. suggest that continued training is important for bilingual teachers [20, 30, 62] and Xiong et al. suggest that institutions should incent professors to take advanced training in teaching methods [62]. Although not the focus of this review, these experience reports highlight the challenges that ELL instructors face when teaching bilingual classes.

With regards to students, authors report that it is important to give students a chance to speak English in class [61] and that understanding English lectures is challenging [59]. They also report that English language skills vary amongst learners [30, 32] and some are less enthusiastic about bilingual teaching [15], however, English is seen as valuable to students' future careers as computer scientists [15, 62]. Liu et al. propose a student-centered approach to bilingual teaching that uses active learning techniques [33].

4.2.2 Translanguaging Eight publications have investigated translanguaging – "the process by which bilingual students and teachers engage in complex discursive practices in order to "make sense" of, and communicate in, multilingual classrooms" [23, p. 299] - in relation to computer science education [37, 49–52, 56–58].

In a theoretical paper, Vogel et al. discuss translanguaging and describe two example pedagogical approaches they used in K-12 settings [58]. They describe computational literacies as the ways that one makes meaning from "computational representations" [58, p. 6], including but not limited to code. Further, they describe translanguaging pedagogies to teach these literacies in a way that "emphasizes what students have and can do, rather than what they lack, or what schools perceive to be the object of their learning (Standard English, for instance)" [58, p. 8]. They argue that translanguaging pedagogy has potential to help us consider "how [computer science] as a community of practice, a discourse, and a literacy evolves" [58, p. 18-19] as participation in computing broadens.

Vogel et al. also used a qualitative analysis technique to investigate students' translanguaging as they learn computational thinking [57] in two middle school English and Spanish dual-language arts classes that integrated computational thinking into other subjects. They collected data from a variety of sources including 50 hours of class observations, audio recordings, and focus groups, and found "computational literacies are intertwined with students' other literacies" [57, p. 3] and "students translanguage to engage in specific [computational thinking] practices." [57, p. 3]

Soosai Raj and colleagues studied translanguaging in the context of instructional designs that combine students' native language and English, as described in Section 4.1.1 [49–52]. Pal discusses translanguaging and code-switching as methods of scaffolding [37].

4.2.3 Content and Language Integrated Learning Content and Language Integrated Learning (CLIL) is an approach that combines language instruction and learning with subject-specific (e.g., computer science) instruction and learning [17]. It has been explored in relation to computing education in five publications [6, 24, 37, 45, 48]. With a CLIL approach to computer science, students' English language skills are further developed through interacting with computing concepts and activities, and, simultaneously, they learn computing concepts via the language that is used in the activities.

4.3 Pedagogical and Curricular Approaches

4.3.1 Culturally Relevant Curriculum Eight publications discuss culturally relevant or culturally responsive curriculum [7, 27–29, 47, 56–58]; all of this work has been published since 2018. As Ryoo et al. state, "curricula and pedagogy need to acknowledge how the CS classroom is not divorced from the larger sociocultural and political contexts within which they sit, and that students should not have to fight to have their voices and perspectives heard within the CS classroom itself." [47, p. 356] Three of these projects were in the context of Research Practice Partnerships (RPP) [27–29, 47, 56–58].

Ryoo et al. describe a RPP that used qualitative data to answer the research question "[f]rom the perspective of minoritized students historically underrepresented in computing, what makes a critical difference in their sense of agency in introductory CS high school classes?" [47, p. 337] They found that it was important to prioritize relevance to students' lives in computer science curricula and pedagogy. As described in Section 4.2.2, in the context of their RPP, Vogel and colleagues have done extensive studies on students' translanguaging while using culturally relevant curriculum [56–58]. Jacob and colleagues describe a RPP aimed at teaching computational thinking to middle school students [27–29]. They integrated culturally relevant computational thinking curriculum with English Language Arts curriculum.

4.3.2 Computational Thinking Computational thinking was a theme in five publications published in 2016 and later [7, 22, 27–29]. The RPP that Jacob and colleagues describe focuses on teaching computational thinking [27–29].

Armenti developed computational thinking and data science curriculum intended to be accessible for English Language Learners and then had a group of expert instructors who design curriculum and assessments for ELL students review it [7]. Her study resulted in curriculum design suggestions of providing multiple opportunities for students to discover meanings of words, using clear, concise language, and allowing students to practice all language modalities (speaking, listening, reading, and writing).

Friss de Kereki and Manataki provide an experience report of their bilingual Spanish-English computing MOOC for high school students that aims to develop computational thinking skills [22].

4.3.3 Universal Design for Learning Four publications [5, 6, 12, 25] mention Universal Design for Learning (UDL), which was created by Meyer and Rose as a set of principles and guidelines for developing curriculum, pedagogies, and assessment strategies that are effective for diverse learners [34]. Burgstahler provides a history and explanation of UDL, general recommendations for using UDL, and examples of how UDL could be applied in a computing course [12]. She describes how using UDL is a *proactive* approach to accessibility while ensuring that students receive university-determined accommodations is a *reactive* approach [12]. UDL does not mandate implementations; instead through its principles and guidelines, one can make decisions about one's own teaching. Burgstahler states that the "potential of [universal design] to improve computing instruction should not be ignored" [12, p. 1]. UDL is not commonly referenced in the computing education literature, but could provide a useful framework for educators who are looking to broaden participation in computing across a variety of populations [25].

Allen et al. [6] describe the application of UDL in curriculum reform of a CS1 course that was taught to students who were English Language Learners. They argue that using UDL will be beneficial to both ELLs and native English speakers.

4.3.4 Teachers' Perceptions and Strategies Three publications have explored instructors' perceptions of teaching computing to English Language Learners independent of a particular pedagogical or curricular focus [24, 43, 54]. Reestman interviewed 10 high-school and university instructors who had taught computing to classes of exclusively (or mostly) ELL students [43]. He found that instructors generally believed that their ELL students were equally able to succeed, but they also described barriers perceived as challenging for ELL students, such as English-language course materials and documentation, and that additional time was necessary for ELL students to complete activities. Reestman's participants described strategies that they used when teaching ELL students, such as speaking the students' native language (if possible), encouraging small group discussions if students have a common, native language, and providing course materials in the native language. Further strategies include providing the information in multiple ways, using analogies, using images, teaching a systematic approach to solving problems, and posing questions to students. Instructors in Reestman's study also mention a variety of institutional, peer, and community supports that are perceived to be helpful to students. Finally, instructors mentioned that they believe many students see English bilingualism as important for their future careers as computer scientists.

Griffith describes an action research project in which eight computer science teachers taught in English using a CLIL approach for the first time [24]. She found that "[w]hat is essential is for professors to see how to use language support strategies with their own contents, with their own students and with their own teaching style." [24, p. 135]. Further, she found that strategies the teachers developed could be transferred to their usual teaching contexts.

Villavicencio et al. report on the first year of a CS4All project in New York City public schools [54]. It is included in this review because one of its goals is increasing access for groups that are underrepresented in computing, including English Language Learners.

It reports teachers' responses to the CS4All professional development workshops but, interestingly, none of the teachers reported any challenges or supports that are specific to ELL students.

5 Tools

Four publications discuss tools and techniques authors have created or used [9, 45, 48, 55]. Banerjee et al. developed BlockStudio, a tool that allows ELL families to learn to program together [9]. They conducted three case studies at community centres that serve ELL families and found that the families were able to engage with the tool in multiple ways. Rimbaud et al. describe preliminary work that used a CLIL approach to create algorithms to supply adaptive, online learning for ELL students [45]. Sisti describes an online computer science English course (CSEC) for undergraduate students who were taking applied computer science [48]. Vishwanathan et al. describe a semi-automated technique to translate deterministic finite automata problems into multiple languages [55].

6 Limitations

This scoping review is limited by a number of factors. We only searched four English-language indexes and databases which returned publications that were written in English. Of the publications that had relevant titles and abstracts, we were unable to locate the full text of three so we had to exclude them at the screening stage.

We conducted this scoping review in a group of two, so all decisions about the indexes and databases to use, the search terms, the inclusion criteria, the screening of publications, the subsequent charting, and the qualitative analysis was influenced by our understandings of the topic, our biases, and our personal subjectivities. All research projects and all forms of literature reviews are influenced by the subjectivities, histories, and biases of the people who undertake them. To mitigate these limitations, we have carefully documented and reported our process in this paper and we have shared the full list of the 182 publications that we considered at https://bit.ly/3yIsfII.

As this is an initial mapping of the literature, we chose to include search terms related to "non-native" English speakers as well as "English Language Learners". Non-native English speakers may be fluent in English, so these two terms are not synonymous. Further research could investigate the nuances of the publications related to these two groups of learners. In most cases, the non-native English speakers who participated in the work reported in this scoping review were also English Language Learners.

7 Conclusions

This scoping review provides a synthesis and summary of the currently available literature about English Language Learners and computer science education. Due to space limitations we chose to focus this paper on presenting the scoping review results; we leave discussion of the implications of these findings for future work.

Despite the relatively small body of work that we found, there has been increasing attention paid to students who are English Language Learners and are learning computer science in English, especially in the past three years. Key themes have been studying what benefits or hinders ELL students, focusing on integrative language skills, and pedagogical and curricular approaches. While the entire body of work is small, the most examined theme has been studying what benefits or hinders ELL students. Work on pedagogical and curricular approaches, particularly on designing culturally relevant curriculum, has emerged in recent years as a key area of focus. We believe that designing culturally relevant curriculum and using translanguaging as a communication strategy are promising avenues of future work. Both avenues are student-focused and take an asset-based approach to students' knowledge, cultural backgrounds, and experiences.

Acknowledgments

We gratefully thank Ann Anderson, Marina Milner-Bolotin, Elisa Baniassad, and Reid Holmes for feedback on the draft of this paper.

References

[1] Abdulwahid Qasem Al Zumor. 2019. Challenges of Using EMI in Teaching and Learning of University Scientific Disciplines: Student Voice. *International Journal of Language Education* 3, 1 (2019), 74 – 90.
[2] Suad Alaofi. 2020. The Impact of English Language on Non-Native English Speaking Students' Performance in Programming Class. In *Proc. ITiCSE 2020*. ACM, Trondheim, Norway, 585–586. https://doi.org/10.1145/3341525.3394008
[3] Ismat Aldmour and Aletta Nylén. 2014. Impact of Cultural and Language Background on Learning Computer Science Concepts. In *Proc. ICTLCE 2014*. IEEE, Kuching, Malaysia, 37–40. https://doi.org/10.1109/LaTiCE.2014.15
[4] Eman Alharbi. 2016. *Characterize the Difficulties that International Computer Science Students Face*. Master's thesis. University of Colorado at Colorado Springs.
[5] Meghan Allen. 2020. Pedagogical and Curricular Practices for Computer Science Education with English Language Learners. In *Proc. ICER 2020*. ACM, Virtual Event, New Zealand, 330–331. https://doi.org/10.1145/3372782.3407108
[6] Meghan Allen, Celina Berg, Jessica Q. Dawson, and Neil Leveridge. 2018. Insights from the Application of Universal Design Principles to Support English Language Learners. In *Proc. WCCCE 2018*. ACM, Victoria, BC, Canada, 1–6. https://doi.org/10.1145/3209635.3209646
[7] Samantha M. Armenti. 2018. *Computer Science Education with English Learners*. Master's thesis. University of Rhode Island.
[8] Mayra Bachrach, Patricia Morreale, and Gail Verdi. 2020. Exploration of Pedagogical Interventions to Improve the Outcomes of Hispanics in AP Computer Science. In *Proc. ISEC 2020*. IEEE, Virtual Event, 1–4. https://doi.org/10.1109/ISEC49744.2020.9280603
[9] Rahul Banerjee, Leanne Liu, Kiley Sobel, Caroline Pitt, Kung Jin Lee, Meng Wang, Sijin Chen, Lydia Davison, Jason C. Yip, Amy J. Ko, and Zoran Popovic. 2018. Empowering Families Facing English Literacy Challenges to Jointly Engage in Computer Programming. In *Proc. CHI 2018*. ACM, Montreal QC, Canada, 1–13. https://doi.org/10.1145/3173574.3174196
[10] Mrwan Ben Idris and Hany Ammar. 2019. The Correlation Between Arabic Student's Proficiency and their Computer Programming Ability at the University Level. *International Journal of Managing Public Sector Information and Communication Technologies* 9, 1 (Oct. 2019), 10. https://doi.org/10.5281/zenodo.3476335
[11] Quincy Brown. 2016. Broadening Participation. *ACM Inroads* 7, 4 (Nov. 2016), 46–48. https://doi.org/10.1145/3008664
[12] Sheryl Burgstahler. 2011. Universal Design: Implications for Computing Education. *ACM TOCE* 11, 3 (Oct. 2011), 1–17. https://doi.org/10.1145/2037276.2037283
[13] J. Elizabeth Casey, Puneet Gill, Lisa Pennington, and Selina V. Mireles. 2018. Lines, roamers, and squares: Oh my! using floor robots to enhance Hispanic students' understanding of programming. *Education and Information Technologies* 23, 4 (July 2018), 1531–1546. https://doi.org/10.1007/s10639-017-9677-z
[14] Ken Chen, Rong Jiang, Qingnian Huang, and Robert M. Randoy. 2011. Practical notes on teaching computer basics in English to international college students in China. In *Proc. ICCSE 2011*. IEEE, Singapore, 660–664. https://doi.org/10.1109/ICCSE.2011.6028725
[15] Ken Chen, Xiaodong Wang, Qingnian Huang, and Paul N. Mekolle. 2012. Bilingual teaching mode for Artificial Intelligent Computer Control: Share. In *Proc. ICCSE 2012*. IEEE, Melbourne, Australia, 1502–1507. https://doi.org/10.1109/ICCSE.2012.6295349
[16] Juliet M. Corbin and Anselm Strauss. 1990. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology* 13, 1 (March 1990), 3–21.
[17] Do Coyle, Philip Hood, and David Marsh. 2010. *Content and Language Integrated Learning*. Ernst Klett Sprachen, Cambridge, UK.
[18] Pedro G. Feijóo-García, Keith McNamara, and Jacob Stuart. 2019. Work in Progress: Native Language Consistency Effect on Learners' Introduction to Block-Based Instructional Technologies. In *Proc. SIGITE 2019*. ACM, Tacoma, WA, USA, 147–147. https://doi.org/10.1145/3349266.3351368
[19] Pedro Guillermo Feijóo-García, Keith McNamara, and Jacob Stuart. 2020. The Effects of Native Language on Block-Based Programming Introduction: A Work

in Progress with Hispanic Population. In *Proc. RESPECT 2020*. IEEE, Portland, Oregon, 1–2. https://doi.org/10.1109/RESPECT49803.2020.9272513
[20] Jian Feng. 2010. Bilingual Teaching of Computer Networks: Research and Practice. In *Proc. ICICCI 2010*. IEEE, Kuala Lumpur, Malaysia, 145–148. https://doi.org/10.1109/ICICCI.2010.121
[21] Jennifer R. Fennema-Bloom. 2010. Code-Scaffolding: A Pedagogic Code-Switching Technique for Bilingual Content Instruction. *Journal of Education* 190, 3 (Oct. 2010), 27–35. https://doi.org/10.1177/002205741019000304
[22] Ines Friss de Kereki and Areti Manataki. 2016. "Code Yourself" and "A Programar": A bilingual MOOC for teaching computer science to teenagers. In *Proc. FIE 2016*. IEEE, Erie, PA, USA, 1–9. https://doi.org/10.1109/FIE.2016.7757569
[23] Ofelia García and Claire E. Sylvan. 2011. Pedagogies and Practices in Multilingual Classrooms: Singularities in Pluralities. *The Modern Language Journal* 95, 3 (Sept. 2011), 385–400. https://doi.org/10.1111/j.1540-4781.2011.01208.x
[24] Mary Griffith. 2017. Tapping into the Intellectual Capital at the University. *Universal Journal of Educational Research* 5 (Jan. 2017), 134–143. https://doi.org/10.13189/ujer.2017.051320
[25] Philip J. Guo. 2018. Non-Native English Speakers Learning Computer Programming: Barriers, Desires, and Design Opportunities. In *Proc. CHI 2018*. ACM, Montreal QC, Canada, 1–14. https://doi.org/10.1145/3173574.3173970
[26] K. James Hartshorn, Norman W. Evans, Jesse Egbert, and Amy Johnson. 2017. Discipline-Specific Reading Expectation and Challenges for ESL Learners in US Universities. *Reading in a Foreign Language* 29, 1 (April 2017), 36–60.
[27] Sharin Jacob, Ha Nguyen, Leiny Garcia, Debra Richardson, and Mark Warschauer. 2020. Teaching Computational Thinking to Multilingual Students through Inquiry-based Learning. In *Proc. RESPECT 2020*. IEEE, Portland, Oregon, 1–8. https://doi.org/10.1109/RESPECT49803.2020.9272487
[28] Sharin Jacob, Ha Nguyen, Debra Richardson, and Mark Warschauer. 2019. Developing a Computational Thinking Curriculum for Multilingual Students: An Experience Report. In *Proc. RESPECT 2019*. IEEE, Minneapolis, MN, USA, 1–2. https://doi.org/10.1109/RESPECT46404.2019.8985944
[29] Sharin Jacob, Ha Nguyen, Colby Tofel-Grehl, Debra Richardson, and Mark Warschauer. 2018. Teaching computational thinking to English learners. *NYS TESOL Journal* 5, 2 (2018), 12–24.
[30] Wenjuan Jiang, Zhaohui Lu, and Shuqian He. 2009. Reflections on the Practice of Bilingual Approach in Teaching Network Security and Management. In *Proc. ICCSE 2009*. IEEE, Nanning, China, 1613–1615. https://doi.org/10.1109/ICCSE.2009.5228310
[31] Wilfred W. F. Lau and Allan H. K. Yuen. 2011. The impact of the medium of instruction: The case of teaching and learning of computer programming. *Education and Information Technologies* 16, 2 (June 2011), 183–201. https://doi.org/10.1007/s10639-009-9118-8
[32] Hanjing Li, Kuanquan Wang, and Yuying Wang. 2009. A Bilingual Teaching Modal in a Programing Language Course. In *Proc. ICETT 2009*. IEEE, Sanya, China, 197–200. https://doi.org/10.1109/ETT.2009.71
[33] Shuang Liu, Xizuo Li, and Li Zuo. 2011. Adopting Communicative Teaching Method in Computer Major Bilingual Teaching. *Theory and Practice in Language Studies* 1, 2 (Feb. 2011), 187–190. https://doi.org/10.4304/tpls.1.2.187-190
[34] Anne Meyer and David H. Rose. 2000. Universal Design for Individual Differences. *Educational Leadership* 58, 3 (2000), 39–43.
[35] Mia Minnes, Bruce Maxwell, Stephanie R. Taylor, and Phillip Barry. 2018. Writing in CS: Why and How?. In *Proc. SIGCSE 2018*. ACM, Baltimore, MD, USA, 402–403. https://doi.org/10.1145/3159450.3159620
[36] Zachary Munn, Micah D. J. Peters, Cindy Stern, Catalin Tufanaru, Alexa McArthur, and Edoardo Aromataris. 2018. Systematic review or scoping review? Guidance for authors when choosing between a systematic or scoping review approach. *BMC Medical Research Methodology* 18, 1 (Dec. 2018), 1–7. https://doi.org/10.1186/s12874-018-0611-x
[37] Yogendra Pal. 2016. *A Framework for Scaffolding to Teach Programming to Vernacular Medium Learners.* Ph.D. Dissertation. IIT Bombay.
[38] Yogendra Pal and Sridhar Iyer. 2012. Comparison of English versus Hindi Medium Students for Programming Abilities Acquired through Video-Based Instruction. In *Proc. 2012 ICTE*. IEEE, Hyderabad, India, 26–30. https://doi.org/10.1109/T4E.2012.30
[39] Yogendra Pal and Sridhar Iyer. 2015. Classroom Versus Screencast for Native Language Learners: Effect of Medium of Instruction on Knowledge of Programming. In *Proc. ITiCSE 2015*. ACM, Vilnius, Lithuania, 290–295. https://doi.org/10.1145/2729094.2742618
[40] Yogendra Pal and Sridhar Iyer. 2015. Effect of Medium of Instruction on Programming Ability Acquired through Screencast. In *Proc. ICLTCE 2015*. IEEE, Taipei, Taiwan, 17–21. https://doi.org/10.1109/LaTiCE.2015.38
[41] Micah D.J. Peters, Christina M. Godfrey, Hanan Khalil, Patricia McInerney, Deborah Parker, and Cassia Baldini Soares. 2015. Guidance for conducting systematic scoping reviews:. *Intl. J. of Evidence-Based Healthcare* 13, 3 (Sept. 2015), 141–146.
[42] Sarah Rauchas, Benjamin Rosman, and George Konidaris. 2006. Language performance at high school and success in first year computer science. In *Proc. SIGCSE 2006*. ACM, Houston, TX, USA, 398–402.
[43] Kyle Reestman. 2019. *Understanding the Relationship Between Native Language and Learning to Program.* Master's thesis. University of Nebraska at Omaha.
[44] Kyle Reestman and Brian Dorn. 2019. Native Language's Effect on Java Compiler Errors. In *Proc. ICER 2019*. ACM, Toronto ON, Canada, 249–257. https://doi.org/10.1145/3291279.3339423
[45] Yann Rimbaud, Tom McEwan, Alistair Lawson, and Sandra Cairncross. 2014. Adaptive learning in computing for non-native speakers. In *Proc. FIE 2014*. IEEE, Madrid, Spain, 1–4. https://doi.org/10.1109/FIE.2014.7044142
[46] A.C Rooij-Peiman. 2020. *The Role of Psychological Capital in First-year Computer Science Students' Retention from a Threshold Concepts Perspective.* Ph.D. Dissertation. Lancaster University.
[47] Jean J. Ryoo, Tiera Tanksley, Cynthia Estrada, and Jane Margolis. 2020. Take space, make space: how students use computer science to disrupt and resist marginalization in schools. *Computer Science Education* 30, 3 (2020), 337–361. https://doi.org/10.1080/08993408.2020.1805284
[48] Flora Sisti. 2012. Online Scientific Language Teaching and Web 2.0. In *Proc. EUROCALL 2012*. Research-publishing.net, Gothenburg, Sweden, 269–273. https://doi.org/10.14705/rpnet.2012.000065
[49] Adalbert Gerald Soosai Raj, Kasama Ketsuriyonk, Jignesh M. Patel, and Richard Halverson. 2017. What Do Students Feel about Learning Programming Using Both English and Their Native Language?. In *Proc. LaTICE 2017*. IEEE, Hong Kong, Hong Kong, 1–8. https://doi.org/10.1109/LaTiCE.2017.8
[50] Adalbert Gerald Soosai Raj, Kasama Ketsuriyonk, Jignesh M. Patel, and Richard Halverson. 2018. Does Native Language Play a Role in Learning a Programming Language?. In *Proc. SIGCSE 2018*. ACM, Baltimore, MD, USA, 417–422. https://doi.org/10.1145/3159450.3159531
[51] Adalbert Gerald Soosai Raj, Eda Zhang, Saswati Mukherjee, Jim Williams, Richard Halverson, and Jignesh M. Patel. 2019. Effect of Native Language on Student Learning and Classroom Interaction in an Operating Systems Course. In *Proc. ITiCSE 2019*. ACM, Aberdeen, Scotland, UK, 499–505. https://doi.org/10.1145/3304221.3319787
[52] Adalbert Gerald Soosai Raj, Hanqi Zhang, Viren Abhyankar, Saswati Mukerjee, Eda Zhang, Jim Williams, Richard Halverson, and Jignesh M. Patel. 2019. Impact of Bilingual CS Education on Student Learning and Engagement in a Data Structures Course. In *Proc. Koli Calling 2019*. ACM, Koli, Finland, 1–10. https://doi.org/10.1145/3364510.3364518
[53] Ashok Kumar Veerasamy and Anna Shillabeer. 2014. Teaching English based programming courses to English learners/non-native speakers of English. *Intl. Proc. of Economics Development and Research* 70 (2014), 1–5. Issue 4.
[54] Adriana Villavicencio, Cheri Fancsali, Wendy Martin, June Mark, Rachel Cole, and New York University. 2018. *Computer Science in New York City: An Early Look at Teacher Training Opportunities and the Landscape of CS Implementation in Schools. Report.* Technical Report. Research Alliance for New York City Schools. https://research.steinhardt.nyu.edu/scmsAdmin/media/users/ks191/CS4All/CS4All_Report.pdf
[55] Aditya Vishwanathan, Mallika Pushpa Bhavatarini, Namratha Ravi, Sneha Umapathi Bhuvaneshwari, Srilalitha Krishnan Murthy, and Viraj Kumar. 2016. An Extensible Multilingual Corpus of DFA Construction Problems. In *Proc. ICCE 2016*. APSCE, Mumbai, India, 78–82.
[56] Sara Vogel. 2020. *Translanguaging About, With, and Through Code and Computing: Emergent Bi/Multilingual Middle Schoolers Forging Computational Literacies.* Ph.D. Dissertation. The City University of New York.
[57] Sara Vogel, Christopher Hoadley, Laura Ascenzi-Moreno, and Kate Menken. 2019. The Role of Translanguaging in Computational Literacies: Documenting Middle School Bilinguals' Practices in Computer Science Integrated Units. In *Proc. SIGCSE 2019*. ACM, Minneapolis, MN, USA, 1164–1170. https://doi.org/10.1145/3287324.3287368
[58] Sara Vogel, Christopher Hoadley, Ana Rebeca Castillo, and Laura Ascenzi-Moreno. 2020. Languages, literacies and literate programming: can we use the latest theories on how bilingual people learn to help us teach computational literacies? *Computer Science Education* 30, 4 (2020), 420–443. https://doi.org/10.1080/08993408.2020.1751525
[59] Xiaohong Wang. 2012. A practice of bilingual teaching in a big-scale class: How we teach programming bilingually in a big class. In *Proc. ICCSE 2012*. IEEE, Melbourne, Australia, 1727–1731. https://doi.org/10.1109/ICCSE.2012.6295399
[60] Xiuyou Wang, Hao Wang, Xuehui Bi, and Jianzhong Fan. 2010. The research and practice on bilingual teaching in computer speciality. In *Proc. ICCSE 2010*. IEEE, Hefei, China, 779–783. https://doi.org/10.1109/ICCSE.2010.5593497
[61] Xiao Mingxia and Ma Xing. 2013. Bilingual teaching of "Signals and Systems"; in Beifang University of Nationalities. In *Proc. ICCSE 2013*. IEEE, Nanning, China, 1199–1203. https://doi.org/10.1109/ICCSE.2013.6554100
[62] Qian Xiong, Jun Peng, and Zhiming Yang. 2010. Study on bilingual teaching of computer science in Chinese universities. In *Proc. ICEIT 2010*, Vol. 2. IEEE, Chongqing, China, V2–226–V2–229. https://doi.org/10.1109/ICEIT.2010.5607559