

Journal of Education and Teaching Learning (JETL)
Volume 8, Issue 3, September 2026
Journal Homepage:
http://pusdikra-publishing.com/index.php/jetl

Content Validity of a Framework for Problem-Solving Competence in Programming Among Information Technology Students: A Two-Round Delphi Study in Vietnam

Nguyen Dinh Tho[1], Nguyen Phuong Chi[2], Pham Tho Hoan[3]
[1,2] *Thu Dau Mot University, Vietnam*
[3] *Hanoi National University of Education, Vietnam*
Corresponding Author : thond@tdmu.edu.vn

ARTICLE INFO
Article history:
Received
10 September2026
Revised
15 October 2026
Accepted
05 November 2026

ABSTRACT

Competence-based education requires that learning outcomes for information technology (IT) programs rest on a competence framework whose content has been validated by experts, yet Vietnam has no framework for problem-solving competence in programming validated through a quantitative, criterion-based procedure. This study validated the content of such a framework, comprising five competence components, twenty-one criteria, and behavioural indicators across four developmental levels, using a two-round Delphi procedure with a purposively heterogeneous, Vietnam-based panel of 23 experts drawn from programming instruction, the software industry, and educational measurement. Round 1 (21 respondents; 91.3%) elicited structured qualitative judgements that generated six documented revisions. Round 2, sent to the same fixed panel (n = 20; 87.0%; 19 also responding in Round 1), rated 169 items against three consensus criteria set in advance: an item-level content validity index (I-CVI) of at least 0.78, a median of at least 4, and an interquartile range of at most 1. Every one of the 169 items satisfied all three criteria simultaneously, the scale-level index (S-CVI/Ave) reached 0.98, and the predefined stopping rule was therefore met after two rounds; a sensitivity analysis restricted to respondents who passed an embedded attention check left this conclusion unchanged. The validated core comprises twenty-one criteria and eighty-four indicators. Two content areas that experts raised without prompting, collaboration with generative artificial intelligence and exception handling, were integrated as twelve extended indicators at the indicator layer rather than as new criteria, preserving the validated criterion structure; these twelve await separate validation. The framework supplies an evidence-based foundation for writing learning outcomes, constructing analytic rubrics, and designing project-based instruction in IT programmes.

Key Word *Competence Framework, Content Validity Index, Delphi Method, Problem Solving in Programming.*

How to cite https://pusdikra-publishing.com/index.php/jetl

This work is licensed under a
Creative Commons Attribution-ShareAlike 4.0 International License

INTRODUCTION

The shift towards competence-based education has changed what a curriculum document in information technology (IT) must be able to demonstrate. Learning outcomes, teaching activities and assessment instruments are expected to trace back to an explicit competence framework, and that framework in turn is expected to carry evidence that its content is defensible. Problem-solving competence in programming (PSCP) occupies the central position in this architecture: international curricular recommendations treat the capacity to move from an ill-defined requirement to a working, evaluated program as the defining competence of a computing graduate (IEEE-CS, 2020). A note on terminology is due here, since our own earlier work rendered this construct as programming problem-solving competence. We adopt the prepositional form deliberately. The compound reading attaches the modifier to the object of solving and so narrows the construct to problems of the programming-exercise type, whereas the framework validated below extends to problems that arise in the course of programming — locating a fault, choosing between library implementations, appraising code an AI tool has proposed — none of which is a programming problem in that narrower sense. The prepositional form also preserves the derivation of the construct, which specialises a general problem-solving competence into a domain rather than positing a self-contained ability; the same structure is used, for example, by Pérez et al (Pérez et al., 2016) in assessing problem-solving competence in computing engineering. What a Vietnamese institution can actually rely on when writing outcomes for its programming courses, however, is far thinner than this consensus suggests.

Our review of the domestic literature identified two persistent gaps. The first concerns evidence. Frameworks currently in circulation are either theoretical proposals that stop short of expert appraisal, or adaptations of international models whose fit with Vietnamese programme structures has never been tested; none reports content validity indices at the criterion or indicator level. Where Delphi studies do exist in Vietnamese educational research, they typically report a single agreement percentage against a threshold chosen without citation, leaving dispersion uncontrolled and offering no scale-level statistic. The second gap concerns currency. Generative artificial intelligence (AI) tools are now routine in the environment where students learn to program, which shifts the assessable locus of competence from producing code towards specifying, reading, appraising and verifying it (Holmes & Miao, 2023; Prather et al., 2023) accordingly recommends that responsible AI use be written into learner

competence standards rather than prohibited. Frameworks built before this shift contain no such component.

The framework validated here was proposed in earlier work on theoretical grounds, drawing on the problem-solving process tradition, design-oriented models of solution generation, and computational thinking (Wing, 2006), and comprising five competence components, twenty criteria and four developmental levels (Thọ et al., 2026). That proposal, like others, carried no empirical warrant. The present study therefore asks two questions. First, does the framework achieve content validity when judged by experts against criteria specified in advance? Second, what structural refinements does expert judgement require, including how newly salient content such as AI collaboration should be positioned? Answering these questions yields three contributions: the first domestically validated PSCP framework carrying quantitative content validity evidence at component, criterion and indicator level; a transparent audit trail linking each between-round revision to the judgement that prompted it; and a defensible mechanism for absorbing emerging content into a validated framework without destabilising its criterion structure.

RESEARCH METHOD

Design and instrument. The study used a two-round Delphi procedure, the iterative technique of eliciting expert judgement through successive rounds with controlled feedback that Dalkey and Helmer (Dalkey & Helmer, 1963) introduced for questions on which no authoritative reference standard exists. Two features of the technique carry the methodological weight here. The first is that experts judge independently, so agreement cannot be produced by the social dynamics of a meeting. The second is that a round is informative only if what counts as agreement, and what would end the procedure, are decided beforehand rather than read off the data afterwards, a requirement Hsu and Sandford (Hsu & Sandford, 2007) identify as the point where Delphi studies most often lose rigour. We therefore fixed both the consensus criteria and the stopping rule before collecting any data. Round 1 elicited structured qualitative judgement through 16 open-ended questions organised in five clusters: competence components, criteria, developmental levels, behavioural indicators, and overall appraisal. Responses were coded thematically, and the resulting revisions produced the instrument rated in Round 2, in which the framework carried 21 criteria, the additional criterion covering the reading and reuse of code from libraries and application programming interfaces having been proposed convergently by several experts. The Round 2 questionnaire contained 169 quantitative items on five-point scales: five components across four facets (relevance, necessity,

comprehensiveness, measurability); 21 criteria across four facets (relevance, clarity, necessity, measurability); four developmental levels across four facets, adding discrimination between adjacent levels and fit with programme structure; and 25 sample behavioural indicators across two facets (discrimination between adjacent levels, observability), drawn from the 84-indicator pool by doubly stratified sampling across criteria and levels. Nine structured questions on outstanding design decisions and one embedded attention check completed the instrument.

Participants. The panel was purposively composed of 23 experts based in Vietnam, drawn from three mutually complementary groups: programming instruction and IT education in Vietnamese higher education institutions (17 experts), software industry recruitment and development (3), and educational measurement and assessment (1). The panel was fixed at 23 members before Round 1 and was deliberately not augmented afterwards, since adding members between rounds would alter the body of judgement that the second round is meant to refine; both questionnaires, together with the Round 1 summary of results, were therefore sent to all 23 members. Round 1 (4–25 May 2026) returned 21 responses (91.3%), comprising 2 associate professors with doctorates, 8 doctorates and 11 masters; 52.4% had more than fifteen years of experience, and 19 of 21 (90.5%) had previously worked on curriculum design, accreditation or competence framework evaluation. Round 2 (3 June – 22 July 2026) returned 20 responses (87.0%). Participation across the two rounds decomposes as follows: 19 members responded in both rounds, 2 responded only in Round 1, 1 responded only in Round 2, and 1 responded in neither. Of the 21 Round 1 respondents, 19 responded again in Round 2, a continuity of 90.5%; the single Round 2 respondent absent from Round 1 is not a late addition to the panel but an original member who did not return the first questionnaire, and who, like every member, received the Round 1 summary of results with the Round 2 instrument and therefore rated it on the same information base as the others.

Linkage across rounds was established by matching each Round 2 identifier against the Round 1 roster on institutional email, then on full name, and in two cases on a personal email whose local part spelled the respondent's full name; the procedure is implemented in the reproducibility script so that readers can re-derive it. One consequence of the response pattern requires explicit statement, because it bears on whose judgement stands behind the Round 2 figures: both Round 1 respondents who did not respond in Round 2 belonged to the industry group. Among the 19 Round 2 respondents whose professional group is documented, the distribution of that round's judgements is therefore 17 in instruction (89.5%), 1 in industry (5.3%) and 1 in measurement (5.3%). This is a

shift in who answered, not a change in panel membership: the two members remained on the fixed panel and received the Round 2 instrument. The professional group of the remaining Round 2 respondent is not captured in either dataset, since that member did not complete the Round 1 demographic section and the Round 2 questionnaire did not repeat it.

This composition is deliberately heterogeneous but unevenly distributed, with instruction accounting for 81.0% of the panel against 14.3% for industry and 4.8% for measurement, and we treat that imbalance as a design feature to be managed rather than concealed. Three mechanisms limit its consequences. Selection within the majority group weighted curriculum and accreditation experience heavily, so that the outcome-writing and feasibility perspectives were represented inside the largest group rather than delegated to the smallest. The three industry experts all hold technical management positions in which they recruit and appraise programmers directly, and their open-ended responses were examined separately when criterion wording was revised, giving their judgement weight disproportionate to their number. Finally, content validity indices were computed across the undivided panel without group weighting, so no consensus conclusion depends on a weighting decision. The thinness of the industry and measurement groups nevertheless remains a limitation, stated as such below. The proportion of master's degree holders, above half the panel, is offset by the seniority and curriculum-development criteria applied at selection.

Analysis. Item-level analysis comprised the mean, standard deviation, median, interquartile range and I-CVI, defined as the proportion of experts rating an item 4 or 5; scale-level content validity was summarised by S-CVI/Ave. An item was treated as having reached consensus only when it satisfied all three predefined criteria simultaneously: I-CVI at least 0.78, the threshold appropriate for panels larger than eight members (Lynn, 1986; Polit & Beck, 2006), a median of at least 4, and an interquartile range of at most 1. The stopping rule, also fixed in advance, required that every item reach consensus and that no new structural proposal emerge. The attention check returned 16 correct responses, 4 blanks and no incorrect answers. The main analysis retained all 20 responses, and three sensitivity analyses reproduced the consensus conclusion without exception: restricting the panel to the 19 continuing experts returned 169 of 169 items in consensus with S-CVI/Ave of 0.979 and a lowest I-CVI of 0.842; restricting it to the 16 responses that passed the attention check returned 169 of 169 with 0.976 and 0.813; and their intersection returned the same result. Agreement therefore does not depend on the one member who responded only in Round 2, nor on the four respondents who left the check blank. Open-ended responses were thematically coded and explicitly traced from Round 1 judgement to Round 2

revision. All quantitative results are reproducible through a base-R script written by the authors.

RESULT AND DISCUSSION

Research results

Round 1 returned strong endorsement of the overall architecture: 90.5% of experts judged the five components comprehensive, 95.2% recommended retaining their sequence, and all respondents affirmed both the four-level progression and the necessity of the criterion set. The substantive value of this round, however, lay in its methodological corrections. Experts asked that internal-state verbs such as "understand" be replaced by observable action verbs; that higher levels be quantified through concrete markers rather than intensifying adverbs; that adjacent levels be distinguished along two explicit axes, the degree of instructor scaffolding and the difficulty of the task; and that developmental level not be tied rigidly to year of study, since a student should be placed by evidence of performance whenever assessment occurs. Two experts independently raised AI without any prompt in the instrument, one arguing that the framework needed openness at the implementation and testing stages to encompass the appraisal, verification and refinement of AI-generated code, the other that the theoretical basis should incorporate work on digital competence and AI. Table 1 traces each principal judgement to the revision it produced and to the Round 2 evidence bearing on it.

Table 1.
Traceability of Round 1 Judgements, Resulting Revisions and Round 2 Outcomes

Round 1 judgement (coded)	Revision entering Round 2	Round 2 outcome
Add reading and reuse of libraries and APIs (2 independent experts)	New criterion TC3.5; confirmation question	80% endorsed; all facets reached consensus
Add exception handling	Dedicated structured question	75% endorsed adding the content
Integrate AI collaboration (2 independent experts)	Dedicated structured question	95% endorsed; integration was the modal form
Remove qualitative adverbs; distinguish levels on two axes	Level descriptors rewritten on two axes	15 of 15 level items reached consensus

Round 1 judgement (coded)	Revision entering Round 2	Round 2 outcome
Use action verbs; quantify upper levels	All 84 indicators rewritten as observable behaviour	50 of 50 sampled items reached consensus; I-CVI 0.95–1.00
List of criterion pairs at risk of overlap	Multiple-choice question to the full panel	One pair confirmed as needing a boundary rule

Source: Authors' Delphi survey data (2026)

In Round 2 every one of the 169 items satisfied all three consensus criteria simultaneously, and S-CVI/Ave reached 0.98 against the 0.90 benchmark recommended by Polit, Beck and Owen (Polit et al., 2007), as Table 2 reports. No item required a third round, so the predefined stopping rule was met. The lowest I-CVI observed anywhere in the framework was 0.85, comfortably above threshold, and its location is informative rather than troubling: the relatively weaker judgements cluster consistently on the measurability facet of the evaluation and optimisation component and on the observability of the Proficient level. Open-ended comments converged on the same reading, namely that higher-order, partly metacognitive competences resist measurement unless the assessment instrument is unusually detailed. Read this way, the pattern is an empirical signal about instrumentation, and it identifies analytic rubric development as the necessary next step rather than exposing a defect in the framework's structure.

Table 2.
Round 2 Consensus Results by Questionnaire Section (n = 20)

Questionnaire section	Items	S-CVI/Ave	Lowest I-CVI	Items reaching consensus
Competence components (5 × 4 facets)	20	0.91–0.98	0.85	20 of 20
Criteria (21 × 4 facets)	84	0.90–1.00	0.85	84 of 84
Developmental levels (4 levels; 4 facets)	15	0.90–1.00	0.85	15 of 15
Sample behavioural indicators (25 × 2 facets)	50	0.98–1.00	0.95	50 of 50
Whole framework	169	0.98	0.85	169 of 169

Source: Authors' Delphi survey data (2026)

The structured questions resolved the outstanding design decisions. The framework was confirmed as applicable across courses and years of study (90% rating 4 or 5); 95% affirmed that it provided a sufficient basis for rubric and learning-outcome development; and overall readiness for implementation averaged 4.15 with 95% at the upper two points. A proposal to merge the criteria covering language syntax and the translation of algorithms into code failed to reach consensus, so both were retained with an explanatory boundary note. Eleven of twenty experts confirmed a risk of overlap between comparing candidate solutions before implementation and evaluating a solution after implementation, which we addressed by adding a decision rule based on the point in the process at which judgement occurs.

Two content areas required a structural decision. Both were endorsed at high rates, exception handling by 75% and AI collaboration by 95%, and for AI the panel divided on form rather than substance, with integration into existing criteria the modal preference (10 of the 19 supporters) against a separate criterion (9 of 19); one further respondent observed that AI-related behaviour appears throughout all five components. We adopted a single integration mechanism for both areas: the 21 validated criteria were left verbatim, and the indicator layer was extended by an eight-indicator AI strand covering four phases of the process, namely specifying requirements for AI, generating and integrating code, testing AI-generated code, and appraisal and optimisation, together with a four-indicator exception-handling strand attached to the implementation and error-correction criteria. Both strands are written as observable behaviour, are levelled on the same two axes, and name no specific tool, so that they do not age with the technology. The criterion covering library and API reuse carries an added scope note stating that the code concerned includes code proposed by AI tools, which leaves its validated wording, and therefore its reported index, intact.

Because the exception-handling decision departs from the literal reading of a 75% endorsement, it warrants an explicit argument. The relevant principle is what counts as a structural unit of this framework: each component and criterion corresponds to a step or cognitive task within the problem-solving process, whereas exception handling is a specific implementation and testing technique, realised through coding behaviour and the design of failure-tolerance mechanisms, and not a cognitive task standing alongside analysis, design, implementation, testing and evaluation. Substantively, the behaviour already falls within the criteria on translating algorithms into code, locating faults and correcting them efficiently; the indicator on designing test cases that cover boundary conditions, exceptions and invalid input was itself validated directly, at I-CVI between 0.95 and 1.00, and three experts stated that the content was

already implicit in the existing criteria. Creating a separate criterion would therefore duplicate three validated criteria and break the rule that each criterion carries exactly four indicators, one per level. We preserved the structure and record openly that this was a research-team decision rather than a direct expression of consensus about form.

Indicator codes remain backward compatible with the validated scheme: *CB[component].[criterion].[level]*, with the suffix a marking AI-strand indicators and e marking exception-handling indicators. CB3.5.3a, for instance, is the level-3 AI indicator attached to the fifth criterion of the third component, implementation, and describes independently checking AI-proposed code against the problem specification before integration. Existing mapping tables and instruments built on the original 84 indicators therefore need no renumbering. The validated core accordingly consists of five components, twenty-one criteria (Table 3) and 84 behavioural indicators, of which 25 were rated directly by the panel under doubly stratified sampling and 59 were validated indirectly through the measurability facet of their parent criteria, while the 12 extended indicators carry no separate index and bring the working framework to 96 indicators.

Table 3.
Structure of the Validated Framework After Two Delphi Rounds

Competence component	Criteria
1. Analysing and understanding the problem	Identifying inputs, outputs and constraints; decomposing the problem into sub-problems; recognising relevant algorithmic patterns and data structures; abstracting the problem.
2. Designing an algorithmic solution	Selecting appropriate data structures; designing logically sound and efficient algorithms; representing algorithms in pseudocode, flowcharts and data structures; analysing and comparing candidate solutions.
3. Implementing the program	Using language syntax correctly; translating algorithms into accurate source code; organising readable source code; using functions and modules effectively; reading and reusing code from libraries and APIs (scope includes code proposed by AI tools).
4. Testing and debugging	Designing test cases; detecting and locating faults; correcting faults efficiently; using debugging tools.
5. Evaluating and optimising the solution	Evaluating solution correctness; analysing algorithmic efficiency; improving and optimising

Competence component	Criteria
	the solution; reflecting on and learning from the process.

Source: Authors' Delphi survey data (2026)

Discussion

The central contribution of this study is the validated core of a PSCP framework for Vietnamese IT students, agreed at a high and remarkably even level across all 169 items submitted for judgement. Its methodological contribution lies in a set of procedural commitments that are uncommon in domestic Delphi reporting: three consensus criteria and a stopping rule fixed in advance, a scale-level statistic alongside item-level indices, an embedded attention check, a sensitivity analysis, and a published audit trail between rounds. Together these allow a reader to re-examine every decision we made rather than accept a summary percentage. That the S-CVI/Ave of 0.98 was obtained on a panel fixed in advance, with 19 of the 21 Round 1 respondents answering again, matters for the same reason: agreement rests on a substantially continuous body of judgement rather than on a reconstituted group, a claim the sensitivity analysis on repeat respondents supports directly. The qualification is that non-response in Round 2 fell on two of the three industry members, so the range of perspectives actually expressed in that round is narrower than the panel's composition implies.

The framework yields three layers of usable instrumentation. At programme level, the criteria and the two-axis level descriptors support learning outcomes written so that attainment can actually be observed. At assessment level, indicators validated at I-CVI 0.95 to 1.00 convert directly into performance descriptors for analytic rubrics, a use that 95% of the panel judged the framework ready to support. At classroom level, the levelled indicators provide the scaffolding gradient for project-based instruction. The AI strand deserves comment in this last respect. Because the need for it was raised by experts before we asked and then endorsed by 95% of the panel, it is best read as a demand arising from teaching practice rather than a researcher's enthusiasm for a topical theme; and because it was integrated at the indicator layer, AI retains its proper status as a context running through the problem-solving process rather than becoming a separate competence that would displace the constructs the framework measures. Practically, this positioning lets an instructor move from policing whether AI was used to assessing how well it was used, for instance by requiring students to submit their own test cases for AI-assisted code, or to justify accepting or rejecting a solution the tool proposed.

Six limitations bound these claims. First, the 12 extended indicators were developed after Round 2 and carry no content validity index, so the study's validity claim applies to the core of 21 criteria and 84 indicators; the extended indicators are being validated through targeted expert consultation and instructor and student survey data. Second, even within the core the strength of evidence is uneven, 25 indicators having been rated directly and 59 validated indirectly through their parent criteria. Third, integrating exception handling at the indicator layer was a research-team decision taken against the literal reading of a 75% endorsement, argued above and disclosed for readers to judge. Fourth, the thin industry and measurement subgroups may narrow the range of perspectives on employer requirements and on the psychometric tractability of indicators, and its effect deepened in Round 2: because both Round 1 respondents who did not answer the second questionnaire came from the industry group, industry judgement in that round came from a single member, so the Round 2 ratings rest even more heavily on the instruction group than the panel's composition suggests. One Round 1 expert also did not receive the indicator table and recorded no answer to three indicator questions, reducing their effective n to 20. Fifth, all panel members were based in Vietnam, so the validity evidence reported here is evidence of fit with Vietnamese programme structures and qualification requirements. The component structure itself derives from international sources and is therefore likely to transfer, but the criterion-level relevance judgements are context-bound; replication with panels in other systems would establish how far the framework travels. Sixth, content validity is not construct validity; establishing the latter requires large-scale student data and factor analysis, which the next phase of this research programme addresses.

CONCLUSION

This study establishes the content validity of the core of a framework for problem-solving competence in programming among information technology students: all 169 judged items reached consensus under three criteria fixed in advance, S-CVI/Ave reached 0.98, and the panel remained 90.5% stable across two rounds. The validated core comprises five competence components, twenty-one criteria and 84 behavioural indicators arranged over four developmental levels, and is, to our knowledge, the first framework of its kind in Vietnam to be published with quantitative content validity evidence at all three of these levels. Beyond the core, twelve extended indicators covering collaboration with generative AI and exception handling were integrated at the indicator layer following the panel's modal preference, a mechanism that absorbs emerging

content without disturbing a validated criterion structure and that may be useful to other framework developers facing the same pressure.

The contribution is therefore both substantive and procedural: a usable framework for writing learning outcomes, building analytic rubrics and designing project-based programming instruction, and a reporting standard for Delphi validation that makes each decision auditable. The limitations identified above set the agenda for what follows. The twelve extended indicators require their own validation evidence; the industry and measurement perspectives warrant a larger share of a future panel; construct validity remains to be established through large-scale student data, factor analysis and rubric-based measurement in authentic instructional settings; and validation with panels outside Vietnam would test whether a structure derived from international sources retains its criterion-level relevance across educational systems.

REFERENCES

Dalkey, N., & Helmer, O. (1963). An experimental application of the Delphi method to the use of experts. *Management Science*, *9*(3), 458–467.

Holmes, W., & Miao, F. (2023). *Guidance for generative AI in education and research*. Unesco Publishing. https://doi.org/10.54675/EWZM9535

Hsu, C.-C., & Sandford, B. A. (2007). The Delphi technique: making sense of consensus. *Practical Assessment, Research, and Evaluation*, *12*(1).

IEEE-CS, A. C. M. &. (2020). *Computing Curricula 2020: Paradigms for Global Computing Education*. Association for Computing Machinery. https://doi.org/10.1145/3467967

Lynn, M. R. (1986). Determination and quantification of content validity. *Nursing Research*, *35*(6), 382–386.

Pérez, J., Vizcarro, C., García, J., Bermúdez, A., & Cobos, R. (2016). Development of procedures to assess problem-solving competence in computing engineering. *IEEE Transactions on Education*, *60*(1), 22–28.

Polit, D. F., & Beck, C. T. (2006). The content validity index: Are you sure you know what's being reported? critique and recommendations. *Research in Nursing & Health*, *29*(5), 489–497. https://doi.org/10.1002/nur.20147

Polit, D. F., Beck, C. T., & Owen, S. V. (2007). Is the CVI an acceptable indicator of content validity? Appraisal and recommendations. *Research in Nursing & Health*, *30*(4), 459–467.

Prather, J., Denny, P., Leinonen, J., Becker, B. A., Albluwi, I., Craig, M., Keuning, H., Kiesler, N., Kohn, T., & Luxton-Reilly, A. (2023). The robots are here: Navigating the generative ai revolution in computing education. In

Proceedings of the 2023 working group reports on innovation and technology in computer science education (pp. 108–159).

Thọ, N. Đ., Chi, N. P., & Hoàn, P. T. (2026). Xây dựng khung năng lực giải quyết vấn đề trong lập trình cho sinh viên ngành Công nghệ thông tin tại Việt Nam: Một tiếp cận tích hợp đa lí thuyết. *Tạp Chí Giáo Dục*, 70–77.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215