Mar 2025

Codebook Definitions – Version 2

These definitions were refined after going through the agreement procedure detailed in agreement_procedure.pdf.

Emboldened text in green are code names/definitions that have edited or added.

~~Struck through text~~ are codes that have been removed.

Name	Description
Barriers to successful debugging	Factors that limit or prevent the likelihood of students having successful debugging experiences
Lack of time for sufficient debugging practice	Mentions of a lack of time for teaching programming, in turn affecting the amount of practice that students get with debugging and more widely, programming.
Unhelpful attitudes	**A lack of interest, confidence, motivation, or pride regarding computing or programming that impedes students' ability or desire to debug.**
Programming-related	**Barriers related to the difficulties of learning to program.**
Conceptual difficulties	**Difficulties with debugging arising from a lack of understanding of certain programming constructs or tooling. This includes mentions of misconceptions and common logical errors.**
Hard to interpret programming error messages	The difficulties of PEMs for students that contribute to difficulty with debugging.
The difficulty with getting syntax exactly right	Mentions of common errors that relate to getting students' syntax, most of which are very minor but prevent a working program nonetheless.

Name	Description
Improving students' debugging ability	**Teaching approaches that teachers employ to help students become more successful debuggers**
Guiding students through debugging strategies	**Teachers modelling or encouraging a particular debugging strategy, often with multiple students.**
Programming strategies to make debugging easier	**Teachers encouraging methods of programming that make debugging less of a difficulty for students**
~~The effect of teacher's debugging strategies~~	~~Mentions of observed changes in students' debugging behaviour, which teachers attribute to their teaching practice around debugging.~~
Utilising PEMs	The view that PEMs are, to some extent, a useful source of automated feedback that can be harnessed in the classroom, as well as mentions of heuristics to "decode" them. A common example is along the lines of "look at the line above the one included in the PEM."
Promoting a positive error culture	Measures in place and morals enforced that promote a positive and healthy perspective on errors, rather than a fear around them.
Celebrating the effort as well as the outcome	Rewarding students who try to solve errors, rather than only focusing on successful resolutions of errors.
Getting used to erroneous code	Deliberately and regularly exposing students to erroneous code that may be created by a student, the teacher, or automatically. The essence of this sub-theme is getting students exposed to errors often, rather than just when they're struggling to program.
Verbal encouragement	**Promoting a positive error culture through positive verbal persuasion around the subject of debugging. Teachers could administer this when students encounter an error, are struggling with debugging, or more generally in their classroom.**

Name	Description
Reliance on the teacher	Discourse related to the common observation that teachers are the first port of call for student debugging, often thought to be caused by a lack of personal resilience. This theme also includes teaching methods that teachers employ to alleviate the pressure on the teacher.
Motivating teacher-independent debugging	Strategies that teachers use that help students to debug independent of the teacher.
Encouraging helpful collaboration	**Enforcing or encouragement students to seek effective support from their peers before asking the teacher for debugging support. Also includes mentions of encouraging effective support that benefits the students collaborating.**
Refusing to help students	Deliberately not providing assistance to students, usually for the purpose of forcing students to attempt debugging by themselves first.
Signposting to other resources for assistance	**Enforcing or encouraging students to explore resources that may help to resolve errors in their programs, for the purpose of fostering teacher-independent debugging. These include the internet, specific websites, cheat sheets, documentation, and large language models.**
Reliance due a lack of resilience	Mentions of the belief that students raise their hand due to a lack of resilience to solve errors without the teacher.
The commonness of teacher reliance	**Mentions of the number of students (either a specific figure or generally) who ask for help from the teacher or who seem reliant on them.**
The toll on the teacher of this reliance	The effect of the heavy teacher reliance on the teacher. These may be challenges that teachers mention, or feelings that this kind of atmosphere evokes in the teachers.
The emotional nature of debugging	A theme regarding the emotional aspect of students' observed debugging experiences, usually described with strong language.

Name	Description
Elation when solving an error	Any observed physical or verbal reactions that indicate strong positive emotions experienced by students at the point of resolving an error.
Reserved reactions for quieter students	Observations of quieter students not reacting much or at all (usually directly compared with visible reactions).
Negative emotion during debugging struggle	Any observed physical or verbal reactions that indicate negative emotions experienced by students, including disengagement with the debugging/programming process.
The resilience of motivated students	Commentary on the positive affect (specifically resilience) that teachers observe among students who are more motivated or confident programmers.
Varying levels of scaffolding for helping students debug	The support mechanisms that teachers provide when they do assist students. These include a range of levels of scaffolding to help them get their programs working.
Asking students questions	Where teachers ask students questions rather than giving answers when providing individual support, usually with the purpose of getting students to "think for themselves".
Encouraging externalisation of ideas	Mentions of where teachers get students to get their thoughts "out of their heads" to help with debugging. This may be through forcing them to explain their code, sharing their thoughts with peers, or encouraging the use of specialist terminology.
Giving students hints	A high level of scaffolding where teachers provide nudges to help students get closer to the answer. This depends on the problem at hand and the student that the teacher is helping.
Giving students the solution	Mentions of teachers literally providing the resolution of the error, usually in the form of a change to make. This doesn't have to always be solicited by the student; a teacher point out the resolution to make without the student asking.