# ACM Task Force on Generative AI and Programming Assessment

Final Report

Steven Gordon
The Ohio State University
Columbus, OH, USA
gordon.1@osu.edu

Paul Denny
University of Auckland
Auckland, New Zealand
paul@cs.auckland.ac.nz

Hieke Keuning
Utrecht University
Utrecht, The Netherlands
h.w.keuning@uu.nl

Natalie Kiesler
Nuremberg Tech
Nuremberg, Germany
natalie.kiesler@th-nuernberg.de

Amruth N. Kumar
Ramapo College of NJ
Mahwah, NJ, USA
amruth@ramapo.edu

Viraj Kumar
Indian Institute of Science
Bengaluru, India
viraj@iisc.ac.in

Juho Leinonen
Aalto University
Espoo, Finland
juho.2.leinonen@aalto.fi

James Prather
Abilene Christian University
Abilene, TX, USA
james.prather@acu.edu

## Abstract

The ACM Education Advisory Committee's Task Force on Generative AI (GenAI) and Programming Assessment was established to understand how GenAI and GenAI coding tools are reshaping programming instruction and assessment and to guide the computing education community through this transition. This report summarizes four connected efforts: (1) a global survey of over 700 educators on policies, concerns, perceived impacts on students' skills, and changes to teaching and assessment; (2) an analysis of concrete course-level adaptations, including shifts toward code comprehension, in-person and oral assessment, and explicit AI literacy; (3) a curated collection of instructor-contributed approaches and tools, made available via a public website; and (4) a hands-on special session at SIGCSE 2026 showcasing exemplar tools and practices. We conclude with key challenges and recommendations for professional development, sharing of best practices, and ongoing community work to integrate GenAI into programming education in ways that support learning, uphold academic integrity, and prepare students for evolving professional practice.

## CCS Concepts

• **Social and professional topics → Computing education**.

## Keywords

Computing Education, Generative AI

## 1 Introduction

Artificial intelligence coding tools are becoming widely used by professional programmers [1, 4]. These tools are also raising critical questions about their impact on student programming skills. AI coding tools can complete current introductory programming exercises and assessments and achieve scores comparable to those of typical beginning students [2]. This raises questions concerning student learning outcomes related to developing programming skills and consequent assessment design, while also ensuring students have sufficient knowledge of the appropriate use of generative AI (GenAI) coding tools [8].

The ACM Education Advisory Committee formed a task force to address several of these emerging issues. In particular, the task force on GenAI and Programming Assessment was formed to track the evolution of alternative approaches to programming instruction with AI coding tools, to document example approaches that can help to guide changes to the curriculum, to trace the potential impacts of the changes on the outcomes provided in the CS2023 ACM/IEEE Computer Science Curricula document[1], and to begin to assemble a sample of best practices that can guide the community through the required changes in programming pedagogy.

As part of this initiative, the task force has undertaken four major tasks:

(1) Conduct a survey to assess the extent and nature of the use of GenAI in teaching programming.
(2) Categorize the various approaches to implementing curriculum changes, resource needs, and barriers to change.
(3) Create a simple tool where faculty can easily share their approaches to integrating GenAI into the curriculum.
(4) Disseminate the results through a report, website, and a workshop or affiliated event at SIGCSE 2026 Technical Symposium.

This report provides a summary of those efforts and offers recommendations for future efforts to address these critical issues.

## 2 Survey of Current Impact and Approaches

### 2.1 Survey overview and distribution

*2.1.1 Survey description.* The survey consists of five major parts. A copy of the full survey is available in the Appendix. The first section focuses on institutional and departmental policies and practices with respect to the use of GenAI. Respondents are also asked for their concerns about student use of GenAI and barriers to integrating GenAI in their courses.

[1] https://csed.acm.org

Section 2 focuses on the possible impacts of GenAI on the range of skills associated with teaching programming. Those include learning coding constructs, program design, implementation, debugging, testing, and documentation.

The next section asks questions about a programming course that was taught in the last 12 months. Along with the course level and number of students, respondents are asked about changes to their teaching and assessment approaches. They are also asked to provide examples of how students have used GenAI in their class and more details on how their assessment methods have changed.

Following a question about the needs for professional development associated with the adaptation of GenAI in courses, we ask for information about the respondents and their institutions. This includes how long they have been teaching, the type of institution, and country.

The final question solicited respondents who might want to share their approaches and tools with the greater community. They were asked to provide their email address if they were willing to participate. This question was purposely detached from the rest of the survey so that the remainder of the survey responses could be anonymous. We then followed up with those indicating a willingness to participate. Details of that effort are described in section 3.1 below.

*2.1.2 Survey distribution.* The first call for survey participants was solicited in early May 2025 through requests by email to a number of professional organizations including SIGCSE members and lists of educators from India, Great Britain, Australia, and the Netherlands. The response to this call was modest, probably due to the fact that many institutions had adjourned for summer break. Accordingly, a second major call was solicited in September 2025. That call included requests to an ACM educator list with 14,000 participants, SIGCSE members, and the other mailing lists.

Responses were accepted until October 1, 2025. At that time we received a total of 763 responses for the survey. The results of that effort are described in detail in the following sections.

## 2.2 Data analysis

The data was collected in anonymous form - any identifying information was collected separately from the survey responses. The survey contained several multiple choice questions, with open-ended responses allowed for the "Other" option. These included barriers (subsubsection 2.3.3), educator concerns (end of subsubsection 2.3.7) and need for professional development (subsubsection 2.3.10). Percentage responses are reported for these questions. Open-ended responses for the "Other" option were manually reviewed and summarized.

The survey also contained several open-ended questions, including on how students are currently using AI (subsubsection 2.3.7), changes to teaching (subsubsection 2.3.8), and changes to assessment (subsubsection 2.3.9). These responses were analyzed using Claude 4.5 Sonnet to find themes. The thematic analysis was inductive in nature, zero-shot (i.e., no a priori examples were provided) and single-prompt (i.e., the prompt statement was crafted once, but was not iteratively revised during interaction with GenAI).

The use of this GenAI approach to find themes in survey responses was aided by the fact that whereas, typically, data points to which thematic analysis is applied are large and ill-structured such as interview transcripts, the responses to a survey question have a clear and limited context (the survey question). Therefore, the data points are short and focused on a single topic. The analysis was conducted in three steps:

*(1) Data Cleaning.* Records with frivolous responses to the question such as "xxx" and "dd", as well as empty responses were deleted. We reviewed the remaining records for mention of any identifying information, and added a unique identifier to each response.

*(2) Prompting.* The prompt statement included: 1) the persona of a computer science education expert to be assumed by GenAI; 2) a description of the task, including the question prompt; and 3) the goal of the analysis. GenAI was asked to: 1) first identify themes in the responses and list them; and 2) for each theme, list the responses in which the theme was found and the words/phrases in each response that led GenAI to identify the theme in the response.

*(3) Validation of Results.* The responses listed for each theme were manually reviewed for correct occurrence of the theme. Any incorrectly classified responses were removed before response counts were reported for each theme.

**Table 1: Respondents by Continent**

| Continent | Count |
|---|---|
| North America | 227 |
| Europe | 106 |
| Asia | 57 |
| South America | 10 |
| Oceania | 9 |
| Africa | 3 |

## 2.3 Survey Results

*2.3.1 Respondents and their demographics.* Of the 763 respondents, 3 did not give consent, 48 did not respond to any questions, and only 198 answered the first question. That left approximately 500 responses to the remainder of the survey with some questions answered by fewer.

Of the 412 respondents who provided their country, most came from North America, next Europe, followed by Asia. Respondents came from 49 countries, with the most from the USA (211), followed by India (33), the UK (20), the Netherlands (17), and Germany (17). Table 1 provides an overview of the respondents by continent.

Of the 431 respondents who stated gender, 113 identified as female, 289 as male, 6 as non-binary or gender diverse, 18 prefered not to disclose, and 5 prefered to self-describe. 442 respondents also characterized their institution. As summarized in Table 2, the vast majority (370 or 77%) work at a university. Few respondents were teachers at secondary school (7), vocational school (3), or primary school (2). Four of the five other institutions were specified as a university granting both bachelor’s and graduate degrees, a research and development unit, a multidisciplinary school, and a professional accreditation body.

Respondents were asked how many years they have been teaching. Of the 246 that answered this question, 22 have been teaching

**Table 2: Institution Type of Respondents**

| Institution Type | Count |
|---|---|
| University (graduate degree granting) | 340 |
| College (bachelor's degree granting) | 62 |
| 2-year college (Associate degree granting) | 23 |
| Secondary school | 7 |
| Vocational school | 3 |
| Primary school | 2 |
| Other | 5 |

5 years or less, 40 between 6 and 10 years, 80 between 10 and 20 years, and 104 more than 20 years.

We further asked whether respondents teach at an institution that serves a minority population in their country. We received 423 responses to that question. Responses were split, with 191 (or 45%) answering "No", 131 (or 31%) selecting "Yes", and 101 (or 24%) selecting "not sure/does not apply in my country".

*2.3.2 Policies on the use of GenAI.* The survey contained two questions about respondents' institutional policies. First, participants were asked whether their institution or department has a policy regarding students' use of GenAI tools. Of the 712 responses, 45% or 320 indicated they do have a policy. 39% or 277 persons responded in the negative to this question, and 16% or 115 respondents were not sure whether their institution or department had a policy on the use of GenAI tools.

Next, we asked to provide a link to those policies or summarize them. 179 respondents submitted a meaningful answer. Unrelated responses were ignored (e.g., "I am not sure", "good", "I have nothing", or links to literature reviews). 95 of the 179 meaningful responses contained hyperlinks to the policy documents of the respondent's institution and/or department. After removing doublets and a link not from a university, policy documents from 81 different institutions remained. A review of those policies showed they all contain useful information for students, staff, and faculty members, and are closely related to academic integrity, ethics, and potential misconduct. Most guidelines make recommendations on which tools to use and in which context this is considered appropriate (i.e., do's and don'ts). Some policy documents further provide templates for acknowledging GenAI tool use in assessments, such as theses and open-book exams, and outline the consequences of not doing so. The hyperlinks we received are, however, just the starting point. Almost all websites contain links to other, more detailed pages for specific target groups, introduce (pedagogical) use cases, or even customized tools. Since we were interested in gaining an overview of policies, we did not analyze all pages and subpages.

Overall, the 179 open-ended responses agree on the following general aspects related to GenAI use and academic integrity:

- Students must fully disclose the use and submission of GenAI contents.
- Students take full responsibility for their submissions as part of a graded assessment.
- Use and submission of GenAI contents is assessed as plagiarism if not disclosed.
- Use and submission of GenAI contents is assessed as plagiarism if not permitted by the faculty or instructor or task.

The following example illustrates these aspects:

> *The policy defines the framework of responsible use of AI in writing theses. AI use must be transparent and approved by the thesis supervisor, all AI use must be specified and described in the thesis. The general guideline is that the student can use AI tools only in some stages of writing the thesis, e.g. generating ideas, but in this case must implement the ideas independently. AI can further be used for translating source documents, grammar and style checks, and text summarization. AI cannot be used to completely replace the students' original work, summarizing texts without indicating the primary sources, for handling sensitive and/or personal data, and in a way that would damage authors' rights.*

At the same time, the responses and documents reflect a certain variety when it comes to specific institutional policies, application contexts, tools, and conditions. For example:

- Any use of AI is allowed as long as reported
- Responsible AI use is encouraged as long as it supports learning (not efficiency)
- Only a specific tool is allowed (e.g., commercial tools, customized/local tool, or tools for which the university purchased a licence)
- Only a specific threshold is allowed (e.g., for 20 or 25% of the work)
- The policy is to let instructors decide on appropriate applications of GenAI tools
- There is no policy on GenAI tools, i.e., it is up to the instructor
- Using GenAI tools is not required due to data privacy concerns and negative impact on learning processes
- GenAI tools are banned by policy, unless instructors explicitly allow them
- GenAI tools are discouraged from being used in programming courses
- GenAI tools can never be used as a source
- GenAI tools are disallowed in CS1 courses
- GenAI tools are generally disallowed in assessments
- Policies are in preparation, but not official/public yet
- There are no policies

*2.3.3 Barriers to integrating AI.* We asked whether respondents have faced any barriers to integrating GenAI in their course. Table 7 shows the selected options from 514 respondents. In total they gave 863 answers, where "Lack of examples of best practices" was mentioned most often by 48% of respondents, followed by "Lack of expertise in GenAI" with 28%, and "Curricular requirements" with 17%. One out of five instructors did not believe there is a need to integrate GenAI at all, and 24% did not face any barriers. Over 100 instructors provided other responses; after removing some invalid responses, we identified themes in the remaining 99 responses. Several of them were closely related to the given options, providing more detailed issues:

*Lack of GenAI expertise and examples of best practices.* Some instructors reiterate lack of expertise on how to integrate GenAI, and the need for best practices and "more success stories". Several instructors mention uncertainty on what to do: "No idea what I'm doing really. Experimenting, trying, playing catchup." Instructors also mention limited knowledge of the technology itself and skills on how to use it: "Instructors don't feel confident in knowing the exact power of the models."

*Curricular requirements.* Some respondents state that there is no space in a course to add new topics and activities, such as introducing a certain tool and discussing how to use it.

*Limitations based on GenAI departmental or institutional policies.* Some respondents provide further details on these barriers. Different forms of resistance are mentioned, from individuals ("scorn from colleagues not ready to make the necessary changes") to faculties and institutions as a whole ("My institution is backward, but I have been free to explore").

Respondents also mention new barriers:

*Access to tools.* A lot of respondents mention problems with accessibility of GenAI tools and their high costs. There are concerns about equal access, and institution not being able to offer access to the right tools. Some instructors also mention not having the means to monitor student use: "I have no access to analytics on assistant use, and cannot evaluate its impact at scale without better tooling."

*Ethical and environmental concerns.* Instructors mention the environmental costs of using GenAI, and data privacy issues. The privacy issue is also mentioned in relation to tool access, not having a tool available that complies with data regulation rules. Self-hosting is mentioned, but often the expenses cannot be covered by the institution.

*Other tool limitations.* Some instructors mention the lack of performance of the tools: "GenAI tools are not stable or predictable. It's difficult to design a curriculum around them."

*Lack of time.* Several instructors mention lack of time to change courses and their assessment. The speed at which AI technologies change is also mentioned as a complicating factor.

*Student behavior and learning benefits.* Student over-reliance on the tools is mentioned as a factor that prevents instructors from integrating GenAI. Some instructors also face students who oppose using GenAI tools ("Some students strongly believe GenAI must be banned from the course"). Instructors mention unclear learning benefits, emphasizing that GenAI undermines their students' learning. One instructor said "I don't think we need to teach it, The tools is pretty much self explanatory." Lastly, several instructors emphasize that GenAI use is not suitable in introductory courses.

*2.3.4 What students are asking about GenAI.* When asked what their students are asking about GenAI tools, a number of concerns are mentioned.

(1) Permission and Policy - is GenAI allowed in class, what counts as cheating, and what the rules are (62 responses).

**Table 3: Barriers that instructors (n=514) have faced with integrating GenAI in their courses. Instructors could select multiple options.**

| Barrier | Count |
|---|---|
| Lack of examples of best practices | 249 (48%) |
| Lack of expertise in GenAI | 146 (28%) |
| Have not faced any barriers | 123 (24%) |
| Do not believe there is a need to integrate GenAI | 101 (20%) |
| Curricular requirements | 88 (17%) |
| Limitations based on GenAI departmental or institutional policies | 53 (10%) |
| *Other* | *103 (20%)* |

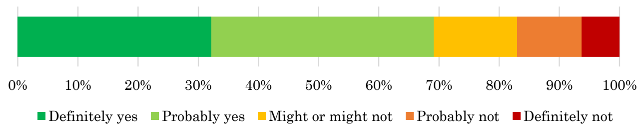


**Figure 1: The extent to which instructors believe the skills to create software have changed because of GenAI.**

(2) Learning and skills - are programming and problem solving skills still necessary if GenAI can generate code (47 responses).
(3) Job market and career impact - what is the impact on employability, relevance of CS degrees, and entry level jobs (54 responses)
(4) Ethics, integrity, and plagiarism - questions about copyright, ethical use of GenAI, and academic honesty (29 responses).
(5) How to use GenAI effectively - learning prompt engineering, documentation, data analysis, and responsible use (31 responses)
(6) There were also a large number (58 responses) that indicated students were being silent. Some interpreted this as a lack of reflection on the part of students about its impacts on their learning. Most, however, interpreted their silence as students using it without questions or students not being concerned at all.

*2.3.5 Impacts of GenAI on software skills.* We asked respondents to indicate whether they believe the skills to create software have changed after the advent of GenAI tools. A total of 512 respondents answered this question (Figure 1). A majority of 69% answered positively to this question, with 32% stating definitely yes, and 37% probably yes. Only 17% of instructors believe that the skills have *not* changed, with 6% saying definitely not, and 11% probably not. The remaining 14% were undecided, thinking the skills might or might not change.

Next, we listed a number of typical tasks related to programming, and asked the respondents to indicate to what extent the way they teach these tasks has been affected by GenAI. Figure 2 shows the results. Teaching small-scale coding, debugging, and documentation are mentioned the most to change "a lot". However, teaching small-scale coding together with program design and learning about

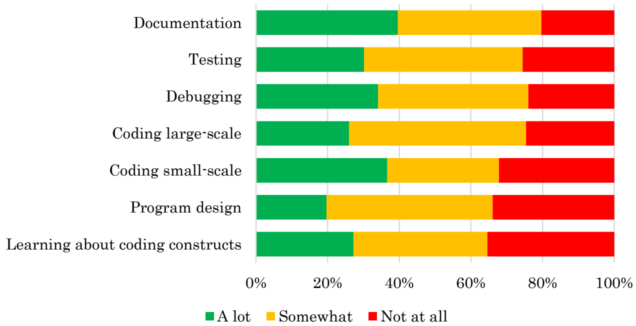


**Figure 2: The extent to which instructors think teaching programming-related tasks have changed.**

**Table 4: Programming Courses Taught**

| Course Category | Count |
|---|---|
| Interdisciplinary / Other | 95 |
| Introductory Programming | 94 |
| Programming Languages & Paradigms | 44 |
| Data Structures & Algorithms | 43 |
| Software Engineering & Development | 28 |
| Data Science & Analytics | 23 |
| Web & Mobile Development | 19 |
| Specialized / Applied Computing | 14 |
| Systems & Architecture | 14 |
| Databases | 11 |
| Capstone / Senior Projects | 9 |

**Table 5: Course Levels**

| Course Level | Count | Percent |
|---|---|---|
| Primary/secondary education (beginner) | 19 | 4 |
| Primary/secondary (intermediate) | 7 | 2 |
| Undergraduate (beginner) | 165 | 36 |
| Undergraduate (intermediate) | 129 | 28 |
| Undergraduate (advanced) | 82 | 18 |
| Graduate / Masters | 62 | 13 |
| **Total** | **464** | **100** |

coding constructs are also the three mentioned most to change "not at all".

*2.3.6 Example courses and their use of GenAI.* Respondents were asked to choose a programming course they taught within the last twelve months. Table 4 shows the categories of courses that were the subject of the responses. The cited courses also include a wide variety by course level and number of students. These are summarized in Tables 5 and 6.

*2.3.7 How students are currently using GenAI.* Respondents were asked to provide some examples of how their students were using GenAI for programming. A thematic analysis of the open-ended

**Table 6: Number of Students in Cited Courses**

| Number of Students | Count | Percent |
|---|---|---|
| 1–10 | 31 | 7 |
| 11–25 | 106 | 23 |
| 26–50 | 117 | 25 |
| 51–100 | 97 | 21 |
| 101–250 | 59 | 13 |
| 250+ | 54 | 12 |
| **Total** | **464** | **100** |

responses provided by 334 respondents using Claude 4.5 Sonnet yielded the following list:

**Abuse of AI:**

- Complete Solution Generation (96 responses) - Students using AI to generate entire programs/assignment solutions
- Academic Dishonesty/Cheating (79 responses) - Explicitly mentioned as cheating or policy violations

**AI for Specific Coding Tasks:**

- Debugging Assistance (68 responses) - Using AI to fix errors, understand error messages, and troubleshoot code
- Code Completion/Snippets (38 responses) - Using AI for auto-completion, small code segments, or specific functions
- Documentation Generation (17 responses) - Creating comments, technical reports, or Javadoc
- Test Case Generation (15 responses) - Creating unit tests or test data
- Code Refactoring/Optimization (8 responses) - Improving existing code quality
- Verification/Checking Work (5 responses) - Confirming correctness of solutions
- Data Processing/Visualization (9 responses) - Creating plots, cleaning data, or processing datasets

**AI as a Learning Aid:**

- Learning/Tutoring Support (62 responses) - Using AI to understand concepts, get explanations, and learn
- Code Explanation/Understanding (27 responses) - Having AI explain what code does
- Syntax/API Help (8 responses) - Looking up how to use specific functions or language features

**AI as a Planning Aid:**

- Prototyping/Starting Points (21 responses) - Using AI to create initial frameworks or templates
- Problem-Solving Guidance (11 responses) - Getting hints or approaches without complete solutions
- Ideation/Brainstorming (6 responses) - Generating ideas for features or approaches

Educators expressed several concerns:

- Skill atrophy: Students producing overly complex code they cannot understand, debug, or explain
- Dependency development: Using AI as the first rather than last resort, bypassing problem-solving practice

**Table 7: Concerns that instructors (n=510) have about students' use of GenAI in the classroom. Instructors could select multiple options. The percentage indicates how many selected the concern.**

| Concern | Count |
|---|---|
| Increased dependency on technology to complete coursework | 444 (87%) |
| Increased cases of cheating or plagiarism | 367 (72%) |
| Misinformation | 269 (53%) |
| Privacy concerns | 142 (28%) |
| *Other* | *126 (25%)* |

- Detection challenges: Difficulty distinguishing AI-generated from student-written code, particularly with integrated IDE tools
- Uneven skill correlation: Less-skilled students tend to use AI more but benefit less, which amplifies the"skills gap"

Innovative educators are:

- Restructuring courses to explicitly incorporate AI, teaching prompting skills and code evaluation
- Requiring students to compare AI-generated solutions with their own
- Implementing oral defenses and in-person assessments
- Creating assignments where AI provides starting points for student customization
- Using AI-generated code as material for debugging and improvement exercises

We asked which concerns instructors have about students' use of GenAI in the classroom (Table 7). With 87% most selected "Increased dependency on technology to complete coursework", followed by "Increased cases of cheating or plagiarism" (72%), and "Misinformation" (53%). A smaller percentage of 28% expressed concerns with privacy, and 126 instructors expressed other concerns.

*2.3.8 Course characteristics and teaching approaches.* The respondents were then asked how the advent of GenAI had impacted their teaching approaches for the course. The vast majority of those who responded indicated that they had made course changes related to the use of GenAI. Of the 469 responses to this question, 64% or 295 indicated that they had instituted course changes.

Those providing descriptions of the changes indicated significant shifts in the approaches to teaching programming. 243 respondents provided open-ended responses describing how they had changed their teaching. The following is the result of thematic analysis of their responses using Claude 4.5 Sonnet, listed along with the number of respondents per change:

**Change in course practices:**

- Increased in-class activities (17 responses) - Increased hands-on, in-class work, live coding, collaborative activities, and interactive learning to engage students directly.
- Assignment design modification (31 responses) - Redesigned assignments to be more AI-resistant, open-ended, creative, or complex projects that are harder for AI to solve directly, or redesigned them to integrate AI use explicitly.
- Demonstrating the use of AI tools in class (18 responses) - Actively demonstrated AI tools during lectures, and live coding sessions, showing both capabilities and limitations, errors, and appropriate usage.

**Change of emphasis in programming:**

- Focus on code comprehension (15 responses) - Shifted emphasis from code writing to code reading, understanding, analyzing, and explaining existing code, including AI-generated code.
- Emphasis on debugging/testing (17 responses) - Increased focus on debugging skills, testing, and verifying AI-generated code for correctness and quality.
- Emphasis on problem-solving/design over syntax (16 responses) - Shifted focus from syntax and coding to higher-level problem-solving, design, algorithms, and conceptual understanding.

**Using AI tools:**

- Teaching/Encouraging AI use/Teaching Prompt Engineering (39 responses) - Either encouraged the use of AI tools or explicitly taught students how to use AI tools effectively, including prompt engineering, when to use AI appropriately, and developing AI literacy skills.
- Requiring documentation (9 responses) - Required students to document, reflect on, and disclose their use of AI tools in assignments.
- Discussion of ethical/responsible use of AI (7 responses) - Incorporated discussions about ethics, responsible use, academic integrity, and the broader implications of AI use.
- Critical evaluation of AI output (11 responses) - Taught students to critically assess, verify, evaluate, and improve generated code.
- Use of custom AI tools (8 responses) - Developed or used custom AI tools, chatbots, or tutors tailored to their courses.

*2.3.9 Changes to assessment.* Respondents were asked if they had made any changes to their assessment methods as a result of GenAI. A total of 459 answered this question with 68% indicating that they had made changes. The types of changes were described by 276 of the respondents. Those open-ended responses were thematically analyzed using Claude 4.5 Sonnet. The following are the changes listed by respondents:

**Where assessments are conducted:**

- Increased In-Person/Proctored Exams (56 responses) - Moving to more in-person exams, supervised, controlled testing environments, paper-based exams, closed-book exams, proctored assessments.
- Reduced Weight on Take-Home Assignments (38 responses) - Decreasing homework/assignment percentage of final grade, devaluing work done outside classroom.
- Elimination of Take-Home Exams (16 responses) - Complete removal of unsupervised exam formats.

**How assessments are conducted:**

- Oral Assessment/Defense (36 responses) - Requiring students to explain their code, assess using oral exams, interviews, and presentations.
- Paper-Based Assessment (35 responses) - Hand-written exams and assignments, pen-and-pencil programming tasks.
- Project-based assessment (34 responses) - using open-ended, real-world and/or group projects that GenAI cannot easily write

**Emphasis of assessments:**

- Focus on Process Over Product (19 responses) - Emphasizing development process, iteration, progress tracking, instead of the finished program.
- Requiring AI Disclosure/Documentation (16 responses) - Asking students to report GenAI usage, including submission of prompts and transcripts.
- Code review/Explanation requirements (15 responses) - Asking students to explain all code, including the code generated by AI
- Debugging/code comprehension focus (12 responses) - students asked more debugging or code execution questions
- More frequent low-stakes assessment (11 responses) - Giving quizzes and tests, often weekly

**Allowing/preventing GenAI use:**

- Technology restrictions (26 responses) - Including disabling GenAI in IDEs, using lockdown browsers, disabling internet access during tests, etc.
- AI detection/monitoring (17 responses) - Checking assignments for AI use, either visually or using tools
- AI-integrated assignments (16 responses) - Using GenAI for assignments is allowed, encouraged or required.

*2.3.10 Need for professional development.* Respondents were asked to respond to the question "What types of professional development would support you to adapt your courses to use GenAI? Select all that apply". The options for the question were the following:

(1) Training on how GenAI works (e.g. prompting strategies)
(2) Training on best practices to integrate GenAI in teaching activities
(3) Training on how to revise assessment when using GenAI
(4) Don't want to adapt my courses
(5) Other (please explain below)

A total of 428 respondents answered this question. 315 (74%) checked "Training on best practices to integrate GenAI in teaching activities", 283 (66%) checked "Training on how to revise assessment when using GenAI", 168 (39%) checked "Training on how GenAI works (e.g. prompting strategies)", 50 (12%) checked "Don't want to adapt my courses" and 79 (18%) checked "Other (please explain below)".

From the 79 open responses, five prominent themes emerged where professional development was requested. The qualitative coding was done by one member of the team. The themes and some representative quotes are below. Some responses, however, did not fall into these themes, e.g., only mentioning that the respondent does not require professional development.

*Assessment redesign and academic integrity (15 responses).*

- "How to verify students' concept mastery in a world where many rely on GenAI for written responses in coursework."
- "complete new types of assessment allowing me to create exams that are on the level of reality. Writing code on paper just to ensure students do not use AI is complete nonsense."
- "Reworking assessments to include GenAI tools without compromising learning outcomes."

*Understanding industry practices and aligning curricula with real-world GenAI use (6 responses).*

- "My program is an applied science program. I need to know more about how employers in my region are using or adapting to AI"
- "It would be useful to know the different attitudes that various software companies have towards GenAI and how they think we should approach GenAI in our teaching."
- "What businesses/employers are actually doing."

*How and what to teach in the GenAI era (22 responses).*

- "Examples of exercises in which the students are supposed to use GenAI."
- "Training on how to continue to teach critical thinking, resilient problem solving, sustained process thinking, and fundamentals in the age of GenAI"
- "How to use GenAI in a beneficial way that keeps the students in control."

*Ethics of using GenAI (12 responses).*

- "It is unclear to me whether integrating these tools for the long run is sustainable (financially and otherwise)"
- "Guidance on how to explain to students how GenAI is affecting their education, as well as the environmental and ethical concerns of using it."
- "We need more training on how to explicitly teach students to develop their own ethics rather than give them more rules to blindly follow."

*Skepticism about integrating GenAI altogether (13 responses).*

- "I think we shouldn't encourage it's use or our reliance on it."
- "all of those good but it isn't really the problem. the problem is misuse. im not sure educating on proper use will ever eliminate the misuse which is rampant in early immature learners."
- "Right now, it's not clear that anyone else knows more. I think there need to be studies done first to show how actual learning can still take place. Right now, it sounds like some want to, by analogy, adapt arithmetic courses to the use of calculators"

## 2.4 Comparisons with prior work

Our survey findings represent the current state of how GenAI is reshaping programming education, including policy updates, observed impacts and assessment changes, but fit alongside a trajectory of similar work taking place over the past few years. In 2023, Lau and Guo found that professors were divided in how they should respond to AI programming tools in the short-term: ban them or embrace them [5]. Although we found that some of our respondents were skeptical about integrating GenAI at all, most seem to now accept that it will play at least some role in computing education moving forward. Professors in Lau and Guo's study also identified some longterm concerns: ethical issues, equity issues, and student learning and over-reliance. These issues are still concerns today and will likely remain so for the immediate future. The professors in Lau and Guo's study also thought that one could successfully embrace AI programming tools through process-based assessments, increased weight of proctored assessment, oral/video assessment, and making AI-proof assessments. Many of these are represented in our data, though it is clear the conversation has shifted in the past three years. For instance, although our respondents said they were modifying assignments to be more AI-resistant by making it harder for AI to solve them directly, there is a distinct shift away from attempting to make "AI-proof" assignments. Finally, Lau and Guo did report that professors in early 2023 thought that code reading and comprehension would become more important and the respondents in our study agree, identifying tasks such as AI-generated code review and debugging.

Prather et al. conducted a working group, also in 2023, that included a survey of instructors' use of generative AI coding tools [8]. They reported that most instructors had not yet observed students using generative AI tools, but the ones who had were seriously concerned with over-reliance and academic misconduct. Three years later, we saw no reports that students were not using AI tools, likely explained by student use now being ubiquitous. Instructors in Prather et al.'s study also identified barriers such as ethical and equity issues that remain relevant today. However, their findings merely encouraged professors to become more proficient in the novel technology and did not provide much prescription for how it could and should be used moving forward.

A second ITiCSE working group by Prather et al. surveyed instructors and specifically focused on those who had already integrated generative AI coding tools into their courses [9]. They reported that instructors had begun to change their teaching practices to adapt to generative AI by focusing on new/emerging skills like prompt writing, a renewed emphasis on classic skills like debugging and code comprehension, explicitly allowing or disallowing generative AI based on the assignment or context, and updating lectures to include generative AI. Instructors interviewed by Prather et al. (2024) also identified how they had changed assessment techniques by putting more weight on proctored exams, focusing on process over product, making AI-resistant assignments, and requiring students to disclose their AI usage. Notably, our respondents largely reported the same ideas, but our results are far more specific and detailed than those of Prather et al. (2024) [9]. This is likely due to our respondents having worked to integrate generative AI into their assessments, lectures, and course policies over multiple semesters. Finally, Prather et al. (2024) also identify ethical and equity issues similar to our respondents. It seems likely that this will remain an important piece of the conversation moving forward, even as instructors "solve" AI integration.

# 3 Tool to Share Current Approaches

## 3.1 Distribution

A final question in the email survey asked respondents if they would be willing to share their experiences with integrating GenAI into their teaching. Those that agreed shared their email addresses. As noted above, that question was detached from the rest of the survey to maintain the anonymity of the survey responses.

Survey results were checked periodically to gather the responses to this question. Each respondent was then sent an individual email with a link to the Google form and a request to share their experiences with the community. For each group of respondents, a second, reminder email was sent approximately a week later.

Altogether 200 of the respondents shared their email addresses. Of those 36 shared their experiences by filling out the Google form. We summarize the nature of their responses in the next section.

## 3.2 Results Summary

All the shared approaches can be found on the task force's website[2].

Across courses and levels, instructors are integrating generative AI into computing education in ways that preserve core learning while leveraging AI for exploration, support, and extension. Many keep traditional programming assignments unchanged but shift a portion of the grade to reflective or creative "+X" work, where students use AI to extend codebases, compare approaches, or document their process. Others weave AI into structured workflows: students generate algorithms or code, manually trace and validate them, refine prompts, and test AI outputs. Course-specific AI tutors, RAG-enhanced assistants, and tools like CodeHelp offer guidance without giving full solutions, helping students debug, study, and practice concepts. In project-based courses, AI serves as a co-developer that helps students navigate libraries, brainstorm designs, or accelerate prototyping.

To ensure students still learn the underlying concepts, instructors increasingly rely on assessments that cannot be outsourced to AI: live code demonstrations, oral exams, in-person mastery checks, paper-and-pencil tests, or code comprehension questions. Many also teach AI literacy explicitly, for example by having students critique AI-generated answers, evaluate AI-proposed code, or discuss ethical and practical limitations. Overall, these approaches aim to help students use AI deliberately and critically, strengthening rather than replacing their computational understanding.

## 3.3 Detailed Analysis

Of the 36 submissions, at least one was out of scope and three similar submissions were made by one person. Hence, there were 33 submissions. Of these, 19 were either described quite poorly, or superficially, or seemed very preliminary. The remaining 14 fell into the following categories:

**3.2.1 Allow AI, but expect more from students (5)**

- +X Report
  - No change to assignment, but 20% of grades given to students who wrote a report on "what else" they did with the assignment's code base
- AI-Augmented Project-Based Learning
  - Transitioned from traditional homework assignments to a more demanding final project that required students to utilize various libraries
  - Students allowed to use AI as a resource to navigate intricate library documentation
  - Student comprehension is evaluated through project live demonstrations
- Offloading tasks after the language is covered
  - After the midway point, integrate the use of offloading simple tasks like having AI create the objects code or HTML/CSS side of the code
- Vibe Coding for Mobile Development

[2]https://acm-education-genai-task-force.github.io/approaches.html

  - ChatGPT used to write code that students integrate into their mobile apps (.NET MAUI using XAML and C#)
  - Encouraged students to think of themselves not merely as coders but as prompt engineers
- Bloom's GenAI Taxonomy for Metacognitive Code Evaluation
  - The pedagogical innovation emphasizes developing metacognitive awareness through comparative analysis, where students systematically examine AI-generated code assessments against their own judgments

**Projects (2)**

- Brainstorming with an LLM
  - Students used an LLM to brainstorm their semester long data science group project
  - Students were provided a simple structure to converse with the LLM
  - Students had to submit their chat as an appendix to their submission
- Project Evaluations
  - In class proctored assessments of the project
  - Students asked detailed questions about their solution e.g., implement one of the functions, test the function, etc.

**"Refute AI" problems on Examinations (2)**

- Model and explain
  - Test questions asking "ChatGPT said this was true, why is it wrong?"
- AI Allowed
  - Paper-and-pencil exam that includes evaluations of code, as if it were proposed by AI and students have to decide whether to accept or not.

**AI tools for learners (5)**

- AutoMCQ
  - Uses GenAI to automatically generate MCQs to check whether students comprehend the code they have written [3]. This is integrated with the CodeRunner automated assessment platform.
- CodeHelp
  - A web application that lets students seek help for programming and conceptual questions in CS [6]. It provides LLM-generated responses that give guidance, explanations, and support without providing complete solutions. Open source: https://codehelp.app/
- Many changes forced by AI availability
  - Integrates an AI tool to an SQL learning game (database course)
  - Paper shows that SQL translations and explanations are good in most cases but common sense and human judgement are still needed from time to time [7].
- AI Tutor utilizing RAG
  - Retrieval Augmented Generation to train the AI Tutor using lecture video transcripts, course materials, and past interactions with students.
  - Student inquiries are handled by the trained AI tutor, while instructors observe the interactions between the AI tutor and students.
- Separate learning from evaluation, use genAI to learn, not to avoid learning
  - Provides a genAI app loaded with the class materials and instructions to help students understand without giving an answer right away

# 4 Symposium

As part of its dissemination activities, the Task Force will host a Special Session at the SIGCSE 2026 Technical Symposium (18–21 February 2026, St Louis, Missouri). The session, titled "Teaching with Generative AI: Tools You Can Use Today", was one of nine accepted special sessions and will involve contributions by presenters from New Zealand, Finland, India, USA, Serbia and Switzerland.

The title and abstract for the special session, as submitted for review, are as follows:

- **Title:** ACM Generative AI Task Force Special Session: Teaching with Generative AI: Tools You Can Use Today
- **Abstract:** This special session, organized by the ACM Task Force on Generative Artificial Intelligence and Student Programming Assessment, will provide a hands-on showcase of novel tools and approaches that are relevant for teaching computing courses in the fast-changing era of generative AI. The tools showcased in the session will be presented by invited educators and researchers, curated by the organizers to highlight diverse approaches, targeting a range of skills including code comprehension, debugging, prompt engineering, assessment, and more. Participants will see live demonstrations of these tools and have the opportunity to try them out on their own devices, gaining practical insights into how they can be integrated into their own teaching. Attendees will leave with concrete ideas and resources to immediately enhance their courses.

## 4.1 Session Overview

The special session was designed to build directly on the ongoing work of the Task Force to document, share, and guide best practices for integrating generative AI into computing curricula. In particular, it extends the community-sharing initiative described in Section 3, in which instructors were invited to contribute examples of how they are incorporating GenAI into teaching and assessment. From the many approaches submitted through that call, the task force reviewed and selected a set of six innovative tools and practices for inclusion in the SIGCSE session. These examples were selected to illustrate a broad spectrum of pedagogical uses of GenAI, from supporting code comprehension to fostering prompt engineering skills and enhancing assessment design.

The session has two primary goals:

(1) to create a forum where educators and researchers can showcase and discuss these novel tools and approaches, and
(2) to provide all participants with hands-on opportunities to explore the tools and consider how they can be immediately adopted in their own teaching practice.

This format emphasizes demonstration and exploration in order to give attendees direct experience with each tool.

### 4.2 Session Structure

The 90-minute session comprises three parts:

*Introduction and Framing (10 minutes).* Members of the Task Force will introduce the session, outline its goals, and present emerging findings from the group's broader investigations into GenAI's impact on programming education. This framing will connect the showcased tools to recurring pedagogical challenges and opportunities identified in the survey and data-gathering efforts.

*Tool Showcases (70 minutes).* Six invited presenters will deliver focused, 7-minute demonstrations of their tools or teaching innovations. These presentations will include an explanation of the pedagogical problem addressed, a live demonstration, and information on how attendees can access and try the approaches or tools. Attendees will be encouraged to bring laptops or devices to experiment alongside presenters. After each demonstration, brief interactive Q&A segments will invite discussion about implementation strategies, barriers, and classroom experiences.

*Closing and Next Steps (5 minutes).* The session will conclude with an invitation for attendees to contribute to the ongoing projects of the taskforce, including updated surveys, resource sharing, and participation in the public repository of approaches. This is designed to ensure that community engagement continues beyond the symposium.

### 4.3 Selected Tools and Presenters

The following six presenters and tools were selected through the community submission process and subsequent review by the task force subcommittee:

- **Kristin Stephens-Martinez** will showcase *Brainstorming with an LLM*, a structured activity where students use iterative dialogue with an LLM to generate and refine research questions for a semester-long data science project.
- **Anastasiia Birillo** will demonstrate *ANVIL*, a system that automatically generates textual and video analogies to explain computing concepts, providing personalized materials to support student engagement and understanding.
- **Sverrir Thorgeirsson** will present an online platform for deploying custom LLM tutors for introductory CS exercises, illustrating its use in a large classroom study and how instructors can adapt it for their own courses.
- **David H. Smith IV** will introduce *Purplex*, an integrated environment that helps novices build code-comprehension and prompting skills through Explain-in-Plain-Language tasks with GenAI-powered automated grading and scaffolding.
- **Mark Liffiton** will demonstrate *CodeHelp*, an automated TA platform featuring "focused tutors" – instructor-defined AI chatbots aligned to specific learning objectives, modules, or assessment goals.
- **Stephen MacNeil** will present *Autocompletion Quiz*, a tool that converts coding tasks into step-by-step quizzes with AI-generated distractors to build expertise around AI suggestions and support students' metacognitive planning and reflection.

### 4.4 Special session outcomes

Following the SIGCSE 2026 special session, the Task Force will integrate key takeaways from the demonstrations and discussions into its upcoming publications and online resources. A summary of showcased tools, pictures from the running of the session, and any participant reflections will be shared through the task force website. The session marks an important step in the work of the taskforce by sharing community practices and strengthening collaboration among computing educators.

## 5 Summary and Conclusions

The survey results reflect a wide range of responses to the advent of GenAI tools and their impact on programming instruction. Instructors are concerned about the impacts of these tools on learning. They are also experimenting with many changes to their teaching and assessment methods. The results can be summarized as follows:

- The advent of GenAI has raised many concerns about student learning and comprehension of computer programming, the potential for cheating, and the dependability and ethics of using GenAI tools. Instructors are concerned about students becoming overly dependent on GenAI tools and fear that students will fail to learn critical programming concepts.
- Many instructors are shifting their teaching strategies from code writing to code comprehension and from syntax to problem solving. These strategies emphasize the use of GenAI as an augmentation to instruction. The tools are used to provide a wider range of examples, to evaluate code, and to provide tutorial support.
- Assessment is shifting away from the grading of individual homework toward larger projects, more complex assignments that GenAI does not yet handle well, oral assessments, and proctored exams. Using GenAI is allowing some instructors to focus on larger scale projects attached to multiple assessments in order to take advantage of GenAI tools while ensuring that the students are developing problem solving and coding skills.
- There is a critical need for instructional examples of effective use of GenAI in programming courses, professional development for course instructors in GenAI tools, and a definition of best practices for integrating GenAI into programming instruction. A substantial percentage of respondents indicated they lacked the expertise in GenAI. They also indicated that the lack of best practice examples is a barrier to integrating GenAI into their teaching.
- There are a significant number of instructors that believe that the use of GenAI will reduce the competency of students, are reluctant to change their teaching approaches, and/or are spending extensive efforts to prevent students from using GenAI tools.

# 6 Next steps

We list some possible next steps in this effort:

- Continue to enhance the website with example programs. We propose continuing to host the website with examples of GenAI teaching and assessment approaches. The additions to the site will be promoted at conferences and through possible cross-references to other GenAI and education sites.
- The ACM should continue to support efforts to integrate GenAI into programming education by sponsoring webinars demonstrating example approaches and panel discussions on changes to assessment approaches. Providing these opportunities for faculty to share their approaches and learn about possible curriculum changes will greatly assist the community with this transition.
- Encourage submissions/publications on GenAI approaches in computing education programs and courses (including experience reports, instructional approaches, assignments, assessments, policies, and rigorous empirical evaluations), and a special session or track on GenAI use at SIGCSE. We envision an annual session where faculty can share their approaches to using GenAI in computing education.
- Efforts should be made to provide professional development workshops for faculty. The survey clearly indicated that a large percentage of current instructors felt that they lacked the expertise to integrate GenAI into their courses.
- Connect with other efforts to integrate GenAI into programming instruction. For example, in the US organizations such as the Computer Research Association, Consortium for Generative AI in CS education, and the Computer Science Teachers Association are all grappling with the same sets of issues. Coordinating efforts with those and other organizations should enhance the potential for success.

# References


[1] Tony Clear, Åsa Cajander, Alison Clear, Roger McDermott, Mats Daniels, Monica Divitini, Matthew Forshaw, Niklas Humble, Maria Kasinidou, Styliani Kleanthous, Can Kultur, Ghazaleh Parvini, Mohammad Polash, and Tingting Zhu. 2025. AI Integration in the IT Professional Workplace: A Scoping Review and Interview Study with Implications for Education and Professional Competencies. In *2024 Working Group Reports on Innovation and Technology in Computer Science Education* (Milan, Italy) *(ITiCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 34–67. doi:10.1145/3689187.3709607

[2] James Finnie-Ansley, Paul Denny, Brett A. Becker, Andrew Luxton-Reilly, and James Prather. 2022. The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. In *Proceedings of the 24th Australasian Computing Education Conference* (Virtual Event, Australia) *(ACE '22)*. Association for Computing Machinery, New York, NY, USA, 10–19. doi:10.1145/3511861.3511863

[3] Martin Goodfellow, Robbie Booth, Andrew Fagan, and Alasdair Lambert. 2025. AutoMCQ – Automatically Generate Code Comprehension Questions using GenAI. arXiv:2505.16430 [cs.SE] https://arxiv.org/abs/2505.16430

[4] Paul Krill. 2024. GitHub survey finds nearly all developers using AI coding tools. *Infoworld*. https://www.infoworld.com/article/3489925/github-survey-finds-nearly-all-developers-using-ai-coding-tools.html Accessed: 2025-11-26.

[5] Sam Lau and Philip Guo. 2023. From "Ban It Till We Understand It" to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1* (Chicago, IL, USA) *(ICER '23)*. Association for Computing Machinery, New York, NY, USA, 106–121. doi:10.1145/3568813.3600138

[6] Mark Liffiton, Brad E Sheese, Jaromir Savelka, and Paul Denny. 2024. CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '23)*. Association for Computing Machinery, New York, NY, USA, Article 8, 11 pages. doi:10.1145/3631802.3631830

[7] Samira Moser and Ela Pustulka. 2025. Enhancing SQL Learning with Large Language Models: Generating Translations and Explanations for Educational Use. In *Proceedings of the Fourth International Conference on Innovations in Computing Research (ICR'25)*, Kevin Daimi and Abeer Alsadoon (Eds.). Springer Nature Switzerland, Cham, 273–282.

[8] James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N. Reeves, and Jaromir Savelka. 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* (Turku, Finland) *(ITiCSE-WGR '23)*. Association for Computing Machinery, New York, NY, USA, 108–159. doi:10.1145/3623762.3633499

[9] James Prather, Juho Leinonen, Natalie Kiesler, Jamie Gorson Benario, Sam Lau, Stephen MacNeil, Narges Norouzi, Simone Opel, Vee Pettit, Leo Porter, Brent N. Reeves, Jaromir Savelka, IV Smith, David H., Sven Strickroth, and Daniel Zingaro. 2025. Beyond the Hype: A Comprehensive Review of Current Trends in Generative AI Research, Teaching Practices, and Tools. In *2024 Working Group Reports on Innovation and Technology in Computer Science Education* (Milan, Italy) *(ITiCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 300–338. doi:10.1145/3689187.3709614

## Appendix: Survey Items

# Student Programming and AI

Start of Block: Default Question Block

Q1 This survey is being conducted as part of a Taskforce of the ACM Education Board on Generative Artificial Intelligence (GenAI) and Student Programming Assessment. We would appreciate your viewpoint on how GenAI tools (ChatGPT, GitHub Copilot, etc.) impact your instructional experiences. The survey should take only 10 - 15 minutes to complete. Participation in the survey is voluntary. Any data collected via this survey will be treated confidentially. Furthermore, any publications resulting from analysis of survey responses will only involve anonymized data, i.e. without data being able to be assigned to a person or institution. We appreciate your participation in the survey. Please see this document to see our privacy practices and provide informed consent: consent_form.pdf

Q2 By checking the box you confirm that you have read the above information and are consenting to the conditions described above and that you are at least 18 years old.

- ○ Yes
- ○ No

Page Break

Q3 Does your institution or department have a policy regarding students' use of GenAI tools?

- ◯ Yes
- ◯ No
- ◯ Not sure

Q4 Please provide a link to those policies or summarize them below?

________________________________________________________________

________________________________________________________________

________________________________________________________________

________________________________________________________________

________________________________________________________________

Q5 Do you allow students to use GenAI in your programming courses?

- ◯ Yes
- ◯ No
- ◯ Partially

Q6 What potential concerns do you have about students' use of GenAI in the classroom? Select all that apply.

- ☐ Privacy concerns
- ☐ Misinformation
- ☐ Increase cases of cheating or plagiarism
- ☐ Increased dependency on technology to complete coursework
- ☐ Other ________________________________________________

Q7 Please elaborate why you do or don't allow GenAI use in your programming courses?

________________________________________________________________

________________________________________________________________

________________________________________________________________

________________________________________________________________

________________________________________________________________

Q8 Have you faced any barriers to integrating GenAI in your course? Select all that apply.

- [ ] Do not believe there is a need to integrate GenAI
- [ ] Limitations based on GenAI departmental or institutional policies
- [ ] Lack of expertise in GenAI
- [ ] Curricular requirements
- [ ] Lack of examples of best practices
- [ ] I have not faced any barriers
- [ ] Other ________________________________________________

Q9 What are your students asking you about GenAI and its impacts on their education?

________________________________________________________________

________________________________________________________________

________________________________________________________________

________________________________________________________________

________________________________________________________________

Q10 Please indicate whether you believe the skills to create software have changed after the advent of GenAI tools

- ◯ Definitely not
- ◯ Probably not
- ◯ Might or might not
- ◯ Probably yes
- ◯ Definitely Yes

Page Break

Q11 For each of the questions below, please indicate to what extent the way you teach these programming tasks has been affected by Gen AI.

Q12 Learning about coding constructs

- ○ Not at all
- ○ Somewhat
- ○ A lot
- ○ Don't know

Q13 Program design

- ○ Not at all
- ○ Somewhat
- ○ A lot
- ○ Don't know

Q14 Coding small-scale: loops, few lines of code

- ○ Not at all
- ○ Somewhat
- ○ A lot
- ○ Don't know

Q15 Coding Large scale: Entire classes, applications

- Not at all
- Somewhat
- A lot
- Dont' know

Q16 Debugging

- Not at all
- Somewhat
- A lot
- Don't know

Q17 Testing

- Not at all
- Somewhat
- A lot
- Don't know

Q18 Documentation

- Not at all
- Somewhat
- A lot
- Don't know

Page Break

Q19 Think of a recent programming course that you teach (or taught within the last 12 months) that is most influenced by GenAI tools and respond to the following questions based on that course.

Q20 What is the name of the course?

________________________________________________________________

Q21 What level is this course?

- ◯ Primary/secondary education beginner
- ◯ Primary/secondary intermediate
- ◯ Undergraduate beginner
- ◯ Undergraduate intermediate
- ◯ Undergraduate advanced
- ◯ Graduate/Masters

Q22 How many students were enrolled in this course?

- ◯ 1-10
- ◯ 11-25
- ◯ 26-50
- ◯ 51-100
- ◯ 101-250
- ◯ 250+

Q23 Have you made changes to your **teaching** approaches in this course as a result of GenAI tools?

- ◯ Yes
- ◯ No

Q24 How did you change your teaching approaches?

________________________________________________________________

________________________________________________________________

________________________________________________________________

________________________________________________________________

________________________________________________________________

Q25 Have you made any changes to the **assessment** approaches you use in this course as a result of GenAI tools?

- ◯ Yes
- ◯ No

Q26 How did you change your assessment approaches?

________________________________________________________________

________________________________________________________________

________________________________________________________________

________________________________________________________________

________________________________________________________________

Q27 Please provide 1-2 examples of how your students are using GenAI for programming.

Q28 If you would like to contribute anecdotes describing how your instruction or assessment of programming courses has changed, we would appreciate it.

Page Break

Q29 What types of professional development would support you to adapt your courses to use GenAI? Select all that apply.

- ▢ Training on how GenAI works (e.g. prompting strategies)
- ▢ Training on best practices to integrate GenAI in teaching activities
- ▢ Training on how to revise assessment when using GenAI
- ▢ Don't want to adapt my courses
- ▢ Other (please explain below) ________________________________________________

Page Break

Q30 Please provide some information about you and your institution.  Note that any data collected via this survey will be treated confidentially and any publications resulting from analysis of survey responses will only involve anonymized data. i.e. without data being able to be assigned to a person or institution.

Q31 For how many years have you been teaching?

________________________________________________________________

Q32 How would you characterize your institution?

- ○ Primary school
- ○ Secondary school
- ○ 2-year college (Associate degree granting)
- ○ Vocational School
- ○ College (bachelor's degree granting)
- ○ University (graduate degree granting)
- ○ Other (please specify below)

__________________________________________________

Q33 Please provide the name of the institution where you are currently teaching (optional)

________________________________________________________________

Q34 Do you teach at an institution that serves a minority population in your country?

- ○ Yes
- ○ No
- ○ Unsure/doesn't apply in my country

---

Q35 Please indicate the country of your institution

________________________________________________________________

---

Q36 What is the gender you identify yourself with?

- ○ Male
- ○ Female
- ○ Non-binary or gender diverse
- ○ Prefer not to disclose
- ○ Prefer to self-describe ________________________________________________

End of Block: Default Question Block