**Generative AI in Computer Science Teaching: 1-Page Cheat Sheet**

**Where Can You Use It?**

|  |  |
| --- | --- |
| **Area** | **Example Uses** |
| **Teaching** | Generate slides, notes, quizzes, code examples, live coding demonstrations. |
| **Learning Support** | AI tutors for instant help, debugging assistance, multilingual explanations, adaptive practice problems. |
| **Assessment** | Automated grading, feedback generation, question banks, plagiarism checks, authentic assessment design. |

**Key Benefits**

* Reduces content preparation time.
* Enables personalized learning pathways.
* Provides 24/7 student support.
* Scales formative assessment easily.

**Main Risks**

* Inaccurate or misleading AI outputs ("hallucinations").
* Student over-reliance on AI-generated answers.
* Equity issues in student access to tools.
* Academic integrity challenges (AI-generated submissions).

**Quick Tips for Lecturers**

* Use GenAI as a **teaching assistant, not a teacher replacement**.
* Emphasize **conceptual understanding** and higher-order thinking.
* Set **clear policies** for acceptable student AI use.
* Design **authentic assessments** that require creativity and critical thinking.
* Stay updated on GenAI capabilities and limitations.

**Useful Tools**

* **ChatGPT / Claude / Gemini:** Explanations, Q&A, content generation.
* **GitHub Copilot / CodeWhisperer:** Code assistance and suggestions.
* **Gradescope / CodeGrade:** Auto-grading and code assessment tools.