CROSS-DISCIPLINARY SKILLS

DEFINITION

Skills are the abilities of a person that can be demonstrated by doing or performing in a relevant context.

Cross-disciplinary skills are domain agnostic abilities that a person can do or perform in a domain specific context.

IN COMPUTING

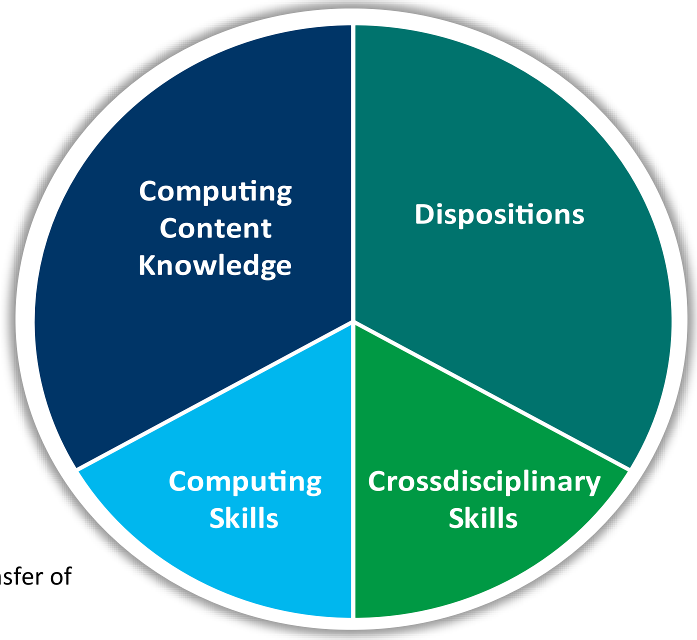
Cross-disciplinary skills can be defined as the abilities valued across different domains that a computing professional can demonstrate by doing or performing in a relevant computing context.

HOW TO IDENTIFY CROSS-DISCIPLINARY SKILLS FOR COMPUTING DISCIPLINES

1. Cross-disciplinary skills are not specific to the computing domain. They can be transferred from one domain to another.
2. They are sensitive to the context and adapt to the context in which they are applied.
3. A computing professional needs to contextualize and adapt these skills to the computing context using relevant content knowledge to successfully navigate work life.

Some examples of Cross-disciplinary skills are:

- Communication
- Problem Solving
- Collaboration/Teamwork
- Lifelong Learning Skills
- Mentorship



PEDAGOGICAL APPROACHES

- Active learning approaches
- Experiential learning approaches
- Multiple opportunities for practice & feedback
- Case studies
- Instructor modeling of the application of skill and explaining their thought process
- Fading scaffolding to transition from a high-level of instructor support to independent work
- Application of skill in multiple contexts to promote transfer of learning across settings

ASSESSMENT APPROACHES

Assessment approaches for skill development measure behaviors directly or indirectly

- Observations of the student applying the skill
- Review of artifacts, such as project reports, presentations, papers, photos/videos of events
- Review of logs, rationale, reflections, etc.
- Peer-review of artifacts, self-assessments, self-and-peer review of team members' interactions

National Science Foundation

This material is based upon work supported by the National Science Foundation under Grant No. 2111097

THEORETICAL PERSPECTIVE

To work and thrive in real-world work environments, professionals not only need domain specific knowledge and skills but also skills that cut across different fields (Hart Research Associates, 2018). Professionals need to demonstrate these skills across various contexts and environments depending on the nature of work.

Recall of learned skills and knowledge occurs best when the learning context replicates the performance context (Imuta et al., 2018). Transfer of skills to different contexts takes place when one engages in handling the learned skills in different ways in different contexts (Georghiades, 2000). Instruction needs to be designed to encourage introspective and reflective thought processes and have learning activities that offer similar and dissimilar contexts in which metacognitive reflection helps learners to apply the learned skills (Smith & Vela, 2001). Thus, along with domain specific knowledge and skills computing educators need to intentionally design instruction to facilitate development and transfer of Cross-disciplinary skills to various contexts.

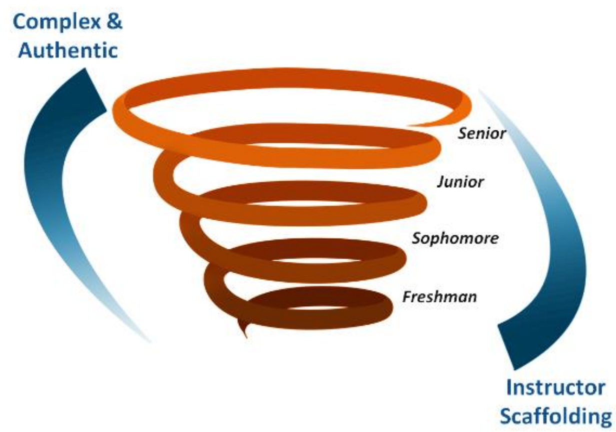
EXAMPLES

Here is an example of a cross-disciplinary skill used in a professional computing context and how it can be taught across multiple courses. A spiral model curricular design will allow students to begin with a large amount of instructor scaffolding and then move on to more authentic, self-directed work.

Professional Task: Documenting Requirements Specifications

Cross-disciplinary Skills: Written Communication

Contextualized Cross-disciplinary Skills: Technical writing



Freshman Year: Experiential learning; high-level of scaffolding; artifact review; review of reflections

In a Freshman year intro to web development course, Professor Brown works with a faculty member from Education to provide a multi-week project-based experience. The web development students partner with education majors to design a website for children. They will first gather requirements from the education majors, and then write a requirements specification document. Because this is their first time, Professor Brown with provide a template with very specific prompts guiding them to include all relevant sections and the construction of each type of requirements statement. After approval from their partners, they will design the UI and code a simple HTML-based site. Computing and education partners will work together with their partners to create the written content of the site. Throughout the course, they reflect on what design decisions they make and why.

Professor Brown uses a rubric to evaluate the artifacts (requirements document and final website) as well as the reflections (indicating that students have made purposeful choices while writing the requirements and the text for the

This material is based upon work supported by the National Science Foundation under Grant No. 2111097

Sophomore Year: Case study; fading scaffolding; peer review

In Professor Nguyen's Sophomore-level course, students will engage in a one-week, online case-based learning activity to provide an early introduction to software engineering and requirements writing in particular. Students select from a number of case studies which include detailed descriptions of a scenario, including the description of a complex, ill-structured problem, people involved, and data that is accessible. These scenarios do *not* include a solution, as it is students' responsibility to propose a design to address the need. A link is provided to a set of requirements standards and examples of well-formed requirements statements. In an online discussion forum, students summarize the case and their tentative approach to the problem, propose a design approach, and write requirements for the high-level design of their solution. Peers give feedback on one another's design approach and requirements (formative peer assessment).

At the end of the week, Professor Nguyen reviews both the original posts and the students' responses. Among other things, Professor Nguyen provides points and feedback related to whether the requirements statements are well-

Junior and Senior years: Experiential learning; no scaffolding; artifact review; peer review

In Junior and Senior years, students engage in authentic projects that include creating requirements documents, design documents, code-embedded documentation, and written reports. Throughout the process, students receive peer-review and instructor-review of their work products (formative assessment). Students are no longer given specific materials to guide how to write each of these – though they are directed to search for their own. The artifacts from each project are included in a student portfolio hosted in GitHub. Faculty review the portfolios, including assessment and feedback on all written components and the write-up of the project within the portfolio page itself. Final feedback and grades are given on the portfolio (summative assessment).

References

Georghiades, P. (2000). Beyond conceptual change learning in science education: focusing on transfer, durability and metacognition. Educational Research, 42(2), 119-139. DOI: 10.1080/001318800363773

Hart Research Associates. (2018, July). Fulfilling the American dream: Liberal education and the future of work. Washington, DC: Author. Retrieved from https://www.aacu.org/sites/default/files/files/LEAP/2018EmployerResearchReport.pdf

Imuta, K., Scarf, D., Carson, S., & Hayne, H. (2018). Children's Learning and Memory of an Interactive Science Lesson: Does the Context Matter? Developmental Psychology, 54(6), 1029-1037.

Smith, S.M., Vela, E. (2001). Environmental context-dependent memory: A review and meta-analysis. Psychonomic Bulletin & Review, 8(2), 203–220.

National Science Foundation

This material is based upon work supported by the National Science Foundation under Grant No. 2111097