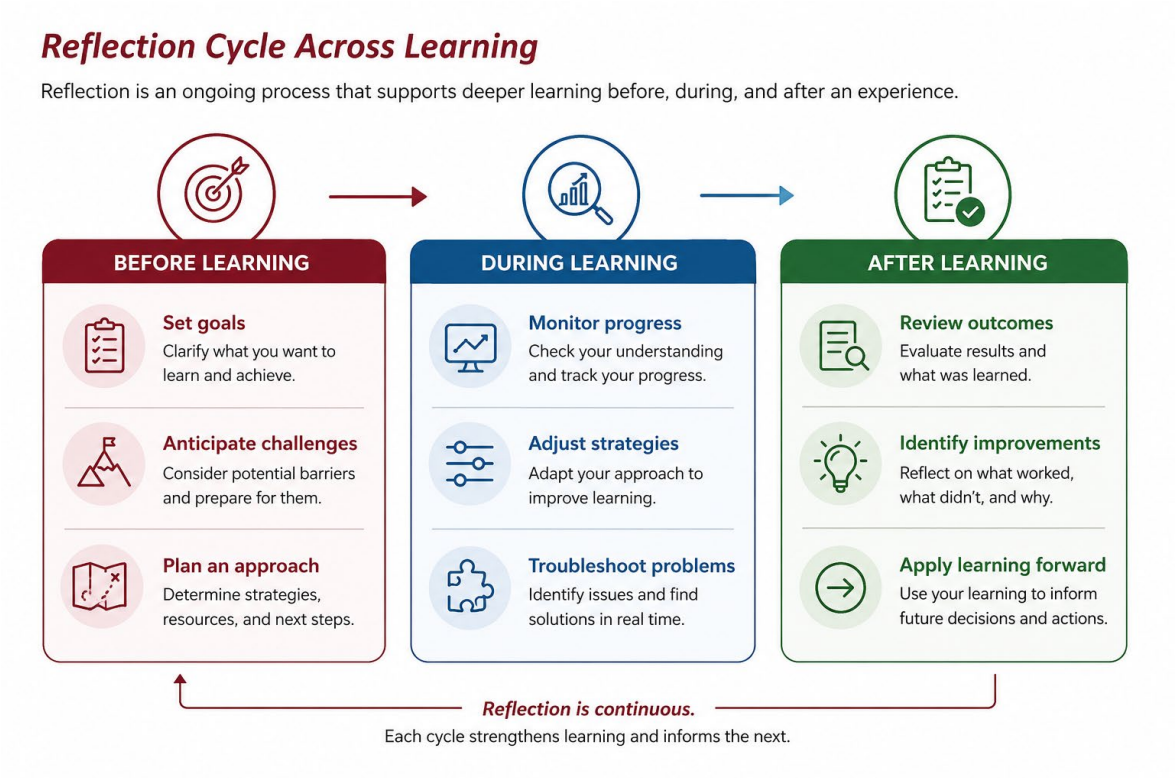
Using Reflection to Support Learning in Computing Courses

Definition

Reflection is the process of consciously thinking about one's learning, decisions, challenges, outcomes, and ways to improve. Reflection is a demanding cognitive activity that is difficult to master (Rodgers, 2002). It can take place before, during, and after a task. When students explain what they are thinking, what is challenging, and how they may adjust their approach, they engage in metacognitive monitoring and make their learning more visible to themselves and others. Reflection goes beyond simply completing an assignment because it asks students to consider how they approach a task, what difficulties they encountered, and how they might improve in the future. For example, when a student encounters a bug while coding, reflection involves not only fixing the error but also identifying what caused the problem, examining the strategy used, and considering how to approach a similar problem differently next time.



In Computing

In computing education, reflection is especially valuable because students are often asked to solve complex problems, make design decisions, debug errors, and work collaboratively. Through reflection, students can develop greater self-awareness and metacognition, transfer what they have learned across courses and tasks, and identify strengths, challenges, and next steps. Reflection can also support important dispositions such as persistence, responsibility, and self-direction. In this way, reflection helps students strengthen technical learning while becoming more aware of how they learn in complex and evolving computing contexts.

Reflection Pedagogical Approach

National Science Foundation

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

Reflection can be embedded into labs, programming assignments, group projects, design reviews, and capstone milestones. Instructors can support student reflection through purposeful prompts, brief check-ins, and regular opportunities for students to think about their learning. Reflection does not need to be long; short prompts can be effective when they are connected to course goals and used consistently. Reflection can be captured through brief written statements, audio or video responses, group discussion, or oral interviews with students. Writing or recording reflection can help students slow down, organize their thinking, and notice how their strategies change over time.

Before an activity

- Ask students to set goals.
- Ask what they expect to be most challenging.
- Ask what strategy they plan to use.
- Ask what prior knowledge or experience may help.

During an activity

- Pause for a brief check-in.
- Ask students what is working well and where they are getting stuck.
- Ask what strategy they are currently using.
- Ask whether they need to adjust their approach.

After an activity

- Ask students what they learned.
- Ask what challenges they faced.
- Ask what worked well in their approach.
- Ask what they would do differently next time.

Practical notes for instructors

- Keep prompts short and purposeful.
- Connect reflection to real course activities.
- Model reflective thinking for students.
- Use reflection regularly, not only at the end.
- Provide brief feedback that encourages deeper thinking.

ASSESSMENT APPROACHES

A simple rubric is often enough for assessing reflection activities (Reinhard et al., 2022). Reflection does not need to be evaluated in a highly formal way. Instead, instructors can focus on the quality of students' thinking and engagement with their learning experiences. Examples of simple reflection rubric criteria include:

- Specificity of examples
- Connection to real experiences
- Self-awareness and insight
- Identification of next steps or improvements
- Responsiveness to feedback

For example, reflections may be evaluated as Emerging, Developing, or Strong based on how thoughtfully students connect experiences to learning and future improvement. Writing quality should not be the primary concern; the depth of reflection matters more. Additional OER materials could also include collections of reflection prompts and adaptable rubric examples for instructors.

THEORETICAL PERSPECTIVE

National Science Foundation

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

Reflection is closely related to metacognition because it helps students become more aware of their own thinking, learning strategies, and areas for improvement (Flavell, 1979). As students explain what they noticed, why they chose a strategy, and how they would change it next time, they engage in metacognitive monitoring. It is also connected to experiential learning, since students learn from concrete experiences and use reflection to improve future performance (Kolb, 1984). In computing education, this makes reflection especially useful for supporting problem-solving, debugging, and continuous improvement across different learning tasks.

EXAMPLES

The activities listed below are examples rather than requirements; instructors may choose one activity or combine several de-pending on course goals, timing, and level.

Introductory Course (1st Year): Building self-awareness and early learning habits

In introductory computing courses, reflection can help students begin noticing how they approach tasks, where they get stuck, and what strategies help them move forward. At this stage, reflection should be brief, structured, and closely tied to specific learning activities.

Scenario 1: Introductory Computing Course

In an introductory programming course, students complete a short lab in which they write and test a simple program. Before beginning, the instructor asks them to identify what they think might be challenging. During the lab, students pause briefly to note where they are getting stuck and what debug-ging strategies they are trying. After the lab, students write a short reflection explaining what worked, what did not, and what they would try differ-ently next time.

Activities

- Short reflection prompts — brief questions completed before, during, and after the lab
- “Muddiest point” responses — students identify the concept they found most confusing
- Pair discussion — students discuss challenges and solutions with a partner
- Instructors debrief — instructor summarizes common challenges and strategies

Learning outcomes

- Early self-monitoring habits — students begin to recognize their own learning process
- Greater awareness of problem-solving strategies — students reflect on different approaches
- Increased confidence in addressing challenges — students build confidence through reflection
- Improved understanding of basic computing concepts — students connect reflection to key concepts

Mid-Level Course (2nd Year): Supporting problem-solving, collaboration, and systems thinking

At the mid-level, reflection can support more complex tasks such as group work, design decisions, and multi-step problem-solving. Reflection at this stage can help students think about both technical choices and how they collaborate with others.

National Science Foundation

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

Scenario 2: Mid-Level Computing or Systems Course

In a mid-level software or systems course, student teams complete a small design or development project. After a milestone submission, students reflect on their contributions, communication challenges, technical decisions, and teamwork experiences. The following activities are examples instructors may choose from based on course needs.

Example Activities

- Team debriefs — short discussions reviewing what worked and what could improve
- Peer feedback — students provide constructive feedback to teammates
- Design reflection reports — brief written reflections on technical and teamwork decisions
- Post-project reflection prompts — guided questions completed after project milestones

Learning Outcomes

- Improved collaboration skills
- Better communication about technical work
- Greater awareness of design trade-offs
- Stronger analytical and reflective thinking

Advanced Applied Course (3rd Year): Applying technical knowledge in realistic and ambiguous environments

In advanced applied courses, reflection can help students work through more open-ended problems that involve uncertainty, multiple possible solutions, and professional judgment. Reflection at this level helps students examine how they adapt their thinking in more complex situations.

Scenario 3: Advanced Applied Computing Course

In an advanced applied computing course, students work on realistic problems requiring analysis, implementation, revision, and adaptation. After completing the task, students reflect on decisions they made, mistakes encountered, and strategies they revised during the process.

Example Activities

- Reflection reports — students explain decisions, revisions, and lessons learned
- Team debrief presentations — groups discuss project outcomes and challenges
- Error analysis — students analyze mistakes and how they addressed them
- Guided post-task reflection — instructor-provided prompts supporting deeper reflection

Learning Outcomes

- Stronger adaptive problem-solving
- Increased situational awareness
- Better understanding of decision-making under uncertainty

Capstone or Advanced Project Course (4th Year): Professional identity and lifelong learning

In capstone and advanced project courses, reflection can help students connect technical work with long-term development, professional judgment, and readiness for future practice. Reflection becomes a way for students to look back on growth across a larger project and identify how they are developing as computing professionals.

National Science Foundation

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

Scenario 4: Capstone or Senior Project Course

In a capstone course, students complete a semester-long project while maintaining a reflective portfolio documenting technical growth, collaboration, leadership, and project development throughout the semester.

Example Activities

- Reflective portfolio — ongoing documentation of learning and project progress
- Capstone retrospective — end-of-project review of successes and challenges
- Professional growth analysis — reflection on skill development and readiness
- Team reflection meetings — collaborative discussions about project experiences

Learning Outcomes

- Professional readiness
- Integration of technical and interpersonal learning
- Greater self-regulation and adaptability

REFERENCES

Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive developmental inquiry. *American Psychologist, 34*(10), 906–911.

Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development.* Prentice Hall.

Reinhard, A., Felleson, A., Turner, P. C., & Green, M. (2022). Assessing the impact of metacognitive postreflection exercises on problem-solving skillfulness. *Physical Review Physics Education Research, 18*(1), 010109.

Rodgers, C. (2002). Defining reflection: Another look at John Dewey and reflective thinking. *Teachers College Record, 104*(4), 842–866.

National Science Foundation

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