CSU-TIPS RESEARCH LESSON STUDY PROPOSAL

2025/26

Adapted from CANMEE Lesson Study Proposal

For the lesson on:	Classes and Object-Oriented Programming (OOP)
Place:	CSULB
Instructor for Research Lesson:	Susan Nachawati (CSULB delivery)
Facilitator:	Kyle Falbo
Lesson Study Team Members:	Kyle Falbo, Kathleen Isenegger, Susan Nachawati, Shabnam Sodagari, Katherine Varela
Commentator:	N/A
Course:	Introductory Computer Science programming course
Title of Lesson:	Connecting People Around the World

Brief Description of Lesson

This lesson introduces classes and objects through a game-based Player object design challenge. Students first connect the idea of a class to an example, then work with partners or groups to determine which attributes, constructors, and methods would be needed for an online game player who can communicate, connect or disconnect, and share biographical information. The lesson uses students' interests in games and real-world communication as a bridge to OOP concepts while inviting multiple valid designs that reflect students' own reasoning and experiences.

1. Students' Competencies and Challenges

Focal students were selected because they were engaging with the course but still continuing to struggle with programming concepts. The descriptions below use anonymized student profiles synthesized from the focal student interview themes.

Focal Student	Competencies and Challenges
Focal Student A Teacher: Susan Nachawati Grade/Level: Undergraduate	Why chosen: Engaged by the course and especially interested in games, but still developing confidence with abstract programming ideas. Assets: Strong game knowledge, curiosity about how games operate, and willingness to connect CS to personal interests. Content Goal: Identify plausible attributes and methods for a Player class and explain why they support the game requirements. Agency Goal: Contribute game-based examples that help the group see the Player object as meaningful and human-centered.

Focal Student	Competencies and Challenges
Focal Student B Teacher: Susan Nachawati Grade/Level: Undergraduate	Why chosen: Can describe programming as a different way of thinking, but may need support translating that metacognitive insight into class structure. Assets: Reflective about learning, able to compare human communication and machine communication, and open to revising thinking. Content Goal: Distinguish between a class, an object instance, a constructor, attributes, and behaviors. Agency Goal: Verbalize design reasoning and ask clarifying questions that make the group's thinking visible.
Focal Student C Teacher: Susan Nachawati Grade/Level: Undergraduate	Why chosen: Continuing to struggle with foundational programming concepts and benefits from stepping back to basics. Assets: Persistence, attention to fundamentals, and willingness to refine thinking through structured steps. Content Goal: Use a class design to instantiate a Player object, modify attributes, and invoke methods. Agency Goal: Use the worksheet and partner discussion to share partial ideas, test them, and revise them without needing a single right answer.

2. Rehumanizing STEM Dimension and Goal

Rehumanizing STEM dimension: Windows and Mirrors, supported by Living Practice.

Rehumanizing STEM Goal: Each student, especially focal students, will see class design as a way to model ideas that are meaningful to them, compare their design choices with peers, and recognize that technical decisions can support human communication, identity, and community.

3. Content Goal

Each student, especially focal students, will understand: That a class is a template for creating objects. They will understand that attributes store information about an object, methods define what an object can do, and constructors set up new objects when they are created.

Each student, especially focal students, will be able to: Recognize and identify class attributes, identify class behaviors or methods, explain the purpose of a constructor, instantiate an object, modify object attributes, and invoke class methods in the context of a Player class.

4. Connecting the Rehumanizing STEM Dimension and the Content Goal

The Windows and Mirrors dimension supports the content goal by making OOP design visible as both personal and comparative. Students use their own experiences with games, online communication, identity, and social interaction as mirrors for deciding what a Player object should store and do. When students compare designs, they use peer solutions as windows into alternative ways to model the same requirements. This helps students understand that OOP is not only syntax; it is a process of abstraction, intentional design, and communication about the real world.

5. Choose the Task and Anticipate Student Responses

Task prompt: "In an online game, players can connect with anyone in the world. Players can: 1) communicate with other players, 2) connect to/disconnect from the game, and 3) get biographical information about themselves and other players. Create a class that represents a Player object that can fulfill these requirements. Clearly identify which functions and attributes would be necessary."

Anticipated student responses: Students may propose attributes such as username, display name, player ID, password, online status, location, avatar, level, health, stamina, bio, friends list, message, and team or guild. They may propose methods such as connect(), disconnect(), sendMessage(), updateMessage(), showChat(), getBio(), updateBio(), addFriend(), removeFriend(), and getStatus().

Anticipated challenges: Students may confuse attributes with methods, design a single specific player instead of a reusable Player class, omit a constructor, struggle to decide which data should be private, or create methods without thinking about parameters and return values. Some students may need support moving from an intuitive game idea to a class skeleton.

6. Unit Plan

Lesson	Unit Learning Goal and Tasks
1	Goal: Prepare students to recognize what information and actions belong together in a program. Task: Review variables, functions, or prior examples, and analyze a simple class as lesson pre-work.
2 - Research Lesson	Goal: Use a human-centered context to design a class with attributes, methods, and a constructor. Task: In partners or groups, design a Player class for an online game and justify which attributes and behaviors are needed.
3	Goal: Translate a conceptual class design into class syntax. Task: Write or revise the Player class header and implementation, including private attributes, public methods, and constructor signature.
4	Goal: Reflect on alternative class designs and extend the model. Task: Compare peer designs, revise the Player class, and answer reflection questions about design choices, abstraction, and audience/user needs.

7. Research Lesson Plan

Indicated time percentages are recommendations; this plan assumes an approximately 50-minute class period and can be adjusted to fit the local course schedule.

Steps, Learning Activities and Teacher Questions	Teacher Support	Formative Assessment
ACT I: IGNITE AND LAUNCH (about 15%) Ignite: Ask students to think about online games, messaging apps, or other social tools they use. Prompt: "When a player connects to an online game, what information does the game need to know about that player, and what actions should the player be able to take?" Expected reactions: Students mention usernames, avatars, online status, chats, friends, health, location, teams, and connecting/disconnecting.	Connect to students' funds of knowledge by validating game and communication examples. Keep the prompt open so students can bring in personal interests. If needed, briefly show a simple Student class as a familiar example of attributes and methods.	Listen for whether focal students volunteer examples, nod, add to a partner's idea, or begin naming data and actions. Note whether students appear motivated by the game context.
Launch Task Present the exact challenge: "In an online game, players can connect with anyone in the world. Players can: 1) communicate with other players, 2) connect to/disconnect from the game, and 3) get biographical information about themselves and other players. Create a class that represents a Player object that can fulfill these requirements. Clearly identify which functions and attributes would be necessary."	Provide a template for requirements, attributes, methods, constructor, and notes. Use partner/group work to reduce the pressure of producing a complete answer alone. Avoid telling students exactly which attributes to choose too early.	Do focal students unpack the requirements? Evidence includes annotated prompts, first lists of attributes and methods, and questions that distinguish what the Player stores from what the Player does.
ACT II: WORK TIME (about 50%) Workshop: Students work with partners or groups to design the Player class. They list attributes, methods, and a constructor, and may write a small class skeleton if time allows. Teacher questions: "What must every Player object know?" "What can a Player object do?" "Which actions need information passed in?" "What should be initialized when a new Player is created?"	Use assessing and advancing questions. Model accountable talk by asking students to explain why an attribute or method is necessary. Position multiple solutions as valid if they meet the requirements. Encourage groups to include design choices that reflect their own understanding of games and online interaction.	Collect evidence from worksheets, student talk, and class design artifacts. Look for attributes/methods aligned with requirements and for students explaining why they made their choices.
Anticipate student solution responses Likely attributes: username, displayName, ID, password, online/connected, message, bio, level, avatar, health, stamina, friends, location, team/guild. Likely methods: connect(), disconnect(), sendMessage(), showChat(), updateMessage(), getBio(), updateBio(), addFriend(), setStatus(), getInfo().	For confusion between attribute and method, ask: "Is this something the player has or something the player does?" For overly specific values, ask: "Would this work for every player, or just one player?" For constructors, ask: "What should a new player have immediately when created?"	Compare anticipated responses to actual student designs. Note which misconceptions appear and how students respond to teacher questions.

Steps, Learning Activities and Teacher Questions	Teacher Support	Formative Assessment
Anticipate focal student exercise of agency Focal Student A may contribute game-specific attributes or methods. Focal Student B may explain the logic behind a design choice. Focal Student C may use the worksheet structure to organize partial ideas. Encourage each form of contribution as computationally meaningful.	Assign competence by naming useful contributions: "That game example helps us think about what a player might need." "Your distinction between data and action helps clarify the class design." Invite focal students to share low-risk pieces, such as one attribute, one method, or one question.	Observe whether focal students contribute ideas, build on peers' ideas, revise their design, or take visible ownership of a part of the solution.
Monitor and sequence During group work, identify one group with a straightforward requirements-based design, one group with a creative game-based design, and one group with a communication or identity-centered design. Select one group's work for whole-class assessment and discussion.	Use sequencing to create windows into different design choices. Handle misconceptions by asking the class to test whether the design meets each requirement rather than labeling it wrong immediately.	Are students explaining, justifying, questioning, and comparing? Do students use evidence from the prompt to evaluate a class design?
ACT III: BRINGING IT ALL TOGETHER (about 35%) Share and Connect: One group shares its Player class. The rest of the class uses the worksheet to answer questions about that group's class: Which items are attributes? Which are methods? What does the constructor initialize? How would you instantiate a Player? How could an attribute be modified? How would a method be invoked?	Position the presenting group as a source of expertise while distributing authority to the class. Ask students to analyze, compare, and justify rather than only copy. Validate different class designs when they satisfy the requirements.	Use worksheet responses as formative assessment. Look for accurate identification of attributes and methods, constructor purpose, and correct examples of instantiation and method invocation.
Summarize Concise summary statement: "A class is a reusable design for objects. Attributes describe what each object knows or stores, methods describe what each object can do, and constructors help create new objects in a valid starting state. In OOP, there can be multiple reasonable designs when choices are intentional and connected to requirements."	Connect the summary back to student work, especially focal student examples and contrasting designs. Make explicit that design choices can reflect personal experience while still needing to satisfy technical requirements.	Check whether students can restate the difference between attributes and methods and identify what a constructor does.
Reflect Exit prompt: "What is one design choice your group made for the Player class, and why did you choose it? What is one idea you learned from another group's design? What part of classes or objects still feels unclear?"	Give students a brief individual reflection moment before collecting responses. Encourage honest uncertainty as useful evidence for the next lesson.	Use exit slips or follow up survey to identify persistent misconceptions and to determine whether students experienced the task as connected to their thinking and interests.

8. Board Plan

The board or projected workspace should reveal the history of the lesson and make student thinking visible. Suggested layout:

Prompt and Requirements	Student Designs: Mirrors	Comparisons: Windows
Post the three game requirements and the question: What does the Player need to know and do? Keep a shared list of nouns and verbs from the prompt.	Record selected group attributes, methods, and constructor choices. Note where choices reflect game experience, communication needs, identity, or user interaction.	Compare two or three designs side by side. Mark similarities, differences, and tradeoffs. Highlight that valid designs can differ when they still meet requirements.

9. Observation Questions

- How do students use their own experiences, interests, or prior knowledge, especially around games, communication, or real-world applications, to make decisions about what belongs in the class?
- How do students respond when the task is partially open-ended? Do they show agency and creativity, or do they need additional structure to feel successful?
- When students compare peer designs, do they recognize multiple valid ways to model the same problem?
- Does instructor guidance support productive struggle without narrowing students' creative design space too early?

10. Reflections

Team reflection: We noticed participation patterns that matter for rehumanizing STEM. Some students were more vocal than others, and observers noted moments where one student appeared more engaged in answering than another. This suggests that future iterations should include more intentional participation structures, such as think-pair-share, written individual responses before group discussion, or rotating group roles, to ensure that more students—especially focal students—have opportunities to contribute publicly and privately.

Students, especially focal students, reflections: Student work and discussion suggest that students were beginning to understand that classes are not just code structures but ways of modeling meaningful objects and interactions. Their ability to name attributes, identify accessor and mutator methods, and respond to questions about constructors and instantiation showed progress toward the lesson's content goals.

Students also showed that collaboration and classroom discourse can support understanding. Although partner work was slow to begin, students became more collaborative over time, and the instructor's movement around the room helped invite questions and check progress. A future student reflection prompt could ask: "How did working with a partner help you decide what attributes and methods belonged in your class?