Objects of Interest: A Lesson Study on Classes

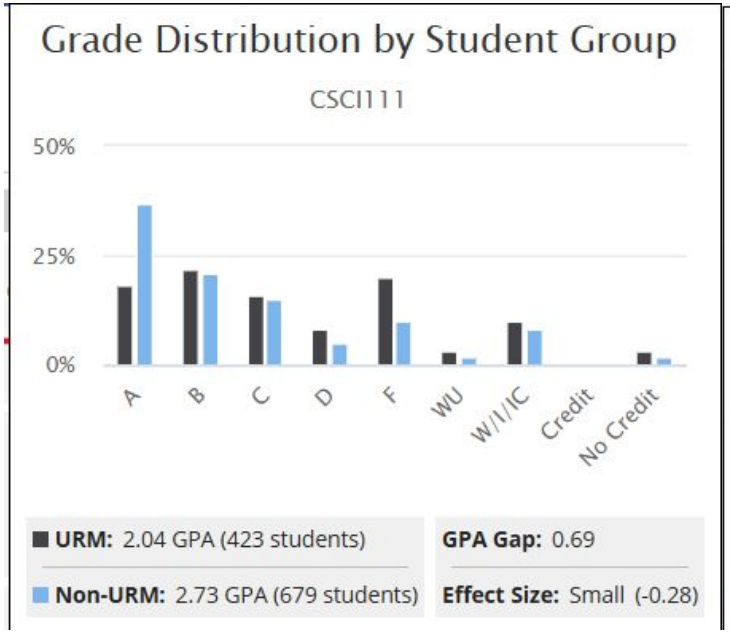
Computer Science
Kyle Falbo, Kathleen Isenegger, Susan Nachawati, Shabnam Sodagari, Katherine Varela

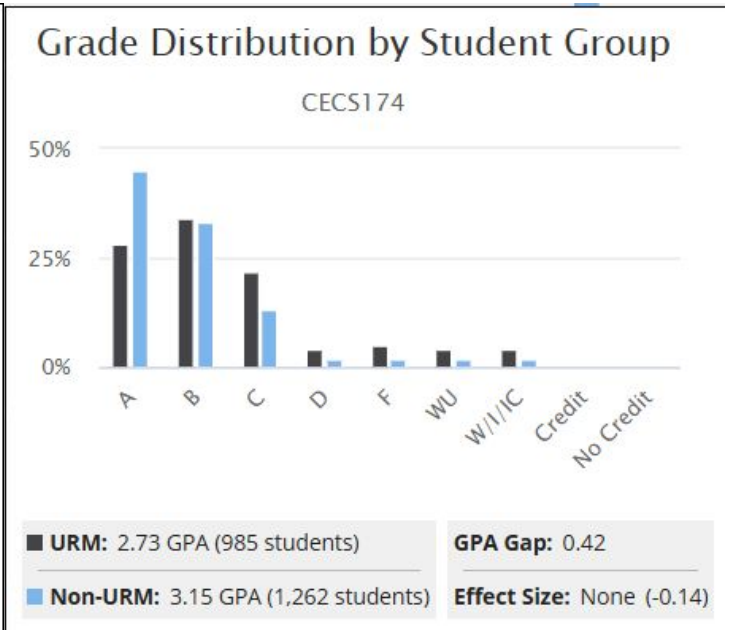
Funding provided through
award number DUE-1953472

Presentation Agenda

- Motivation
- Interviews with Focal Students
  - Focal Student Responses
- Rehumanizing STEM Dimension
- Content Focus
- Delivering Lesson
- Reflections

Motivation





Experience and survey data show that students struggle to understand **classes and objects** in early programming courses.

CSU Course GPA Equity Gaps Dashboard shows a noticeable GPA gap for students from underrepresented minority groups in courses covering these topics.

Interviews with Focal Students

- Group interviews conducted lasting ~1 hour
  - Focal students were identified as those engaging with the course, but still continuing to struggle
- Our goal was to better understand how focal students…
  - Engage/understand/perceive programming as a field of study
  - Engage/understand their own learning
  - Culturally relate to computer science topics

Focal Student Responses

What students say about…

Programming as a field of study…

“Programming is a just a way of thinking differently.”

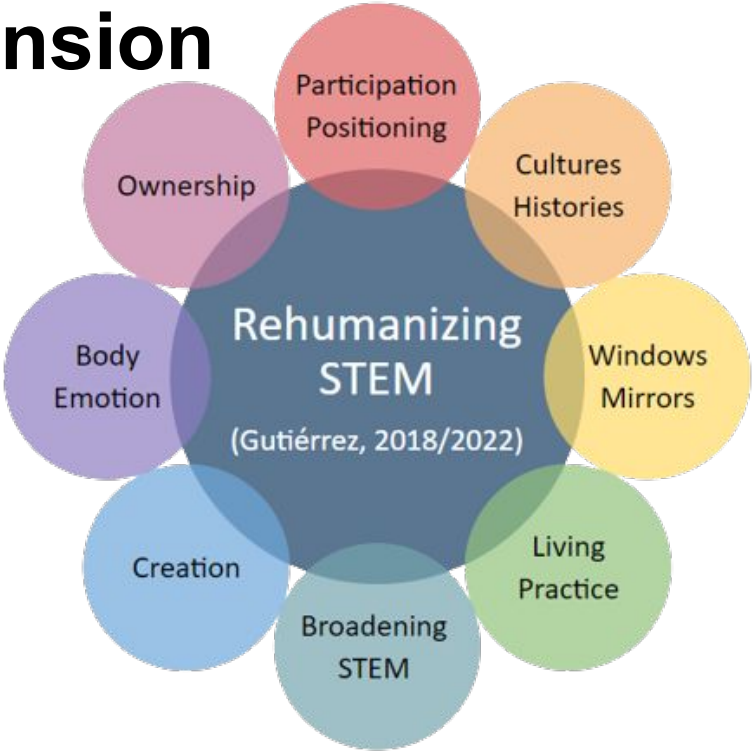
How they engage their own learning…

“What makes it challenging is how different you have to think. You are not communicating with another person but instead a machine...I have to take a step back and look at the basics to try and refine my thinking.”

How they culturally relate to computer science topics…

“I feel like I prefer problems that relate to games. I like to figure out how games are coded. It is interesting to see how games operate.”

Rehumanizing STEM Dimension



Windows & Mirrors

We wanted to create space for students to develop solutions in ways that resonate with their own thinking and experiences. At the same time, we structured opportunities for students to compare and learn from how their peers approached the same prompt. We also aimed to normalize reflective practices that connect coding solutions to students' identities and communities.

Living Practices & Futures

Another goal of our lesson study was to encourage students to brainstorm how they would like to design tools that are practical and accessible to a wide variety of users. Students were encouraged to think how their design could be used to facilitate human communication and interactions regardless of the user's background or geographic location.

Content Focus

In introducing **Classes and Object Oriented Programming**:

- We can frame students' code design as **reflection** of their **own thinking** (**mirrors**) while exposing them to **alternative design choices** from their peers or other designers (**windows**).

- We can also honor *living practice* by helping students see **coding as an extension** of **their daily experiences**, cultural backgrounds, and personal ways of making sense of the world.

Rehumanizing STEM: An OOP-Driven Perspective

Rehumanizing STEM is about bringing **human experience, identity, creativity, and community** back into science and engineering education. Rather than allowing STEM to feel rigid, abstract, or detached from lived realities, a rehumanized approach emphasizes:

- **Narratives, stories, and lived experience**
- **Inclusive and collaborative learning**
- **Authentic problem-solving connected to real communities**
- **Student agency, identity, and belonging**

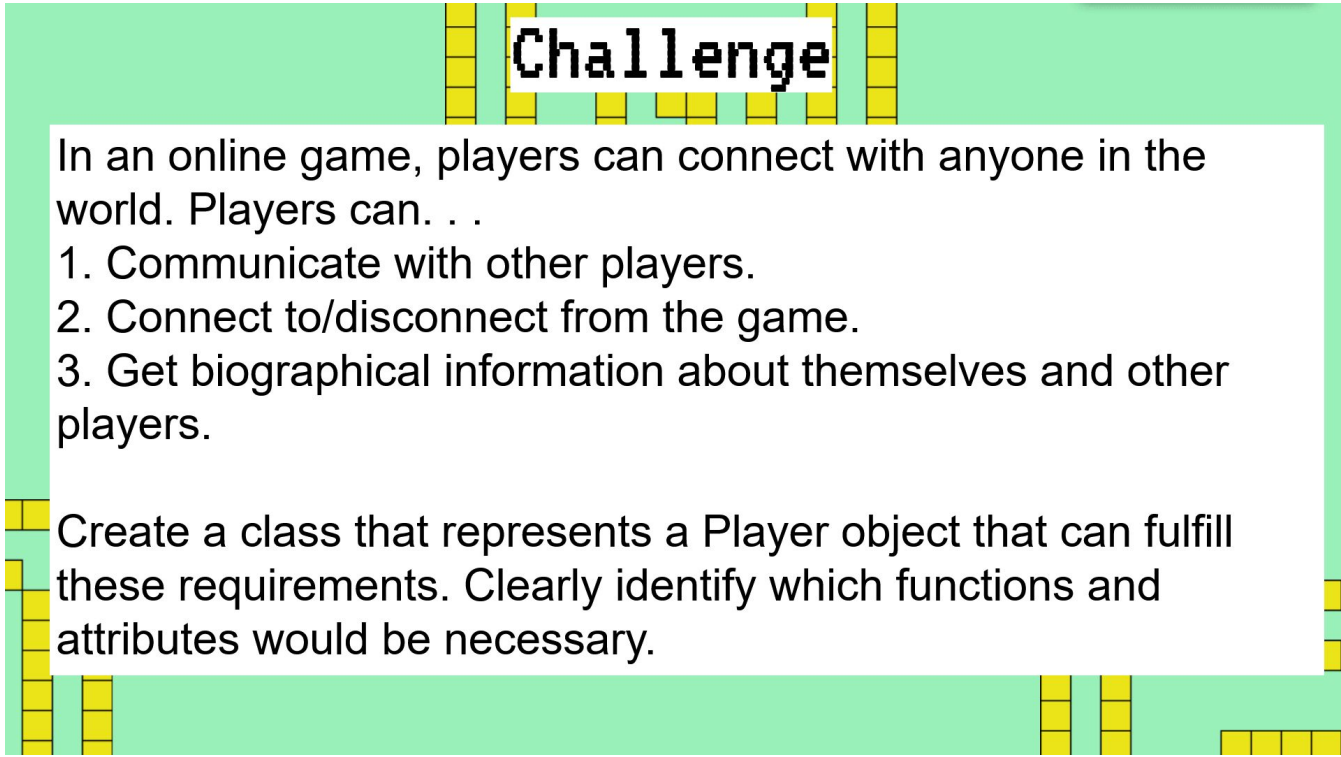
Lesson Pre-Work: Example Class

student.h > ...

```
1   #ifndef STUDENT_H
2   #define STUDENT_H
3   #include <string>
4   using namespace std;
5   class Student{
6       private:
7           string name;
8           int ID;
9           int graduationYear;
10
11      public:
12          Student(string s_name, int s_ID, int s_year);
13          void setGradYear(int year);
14          int getGradYear();
15  };
16  #endif
```

Delivering Lesson

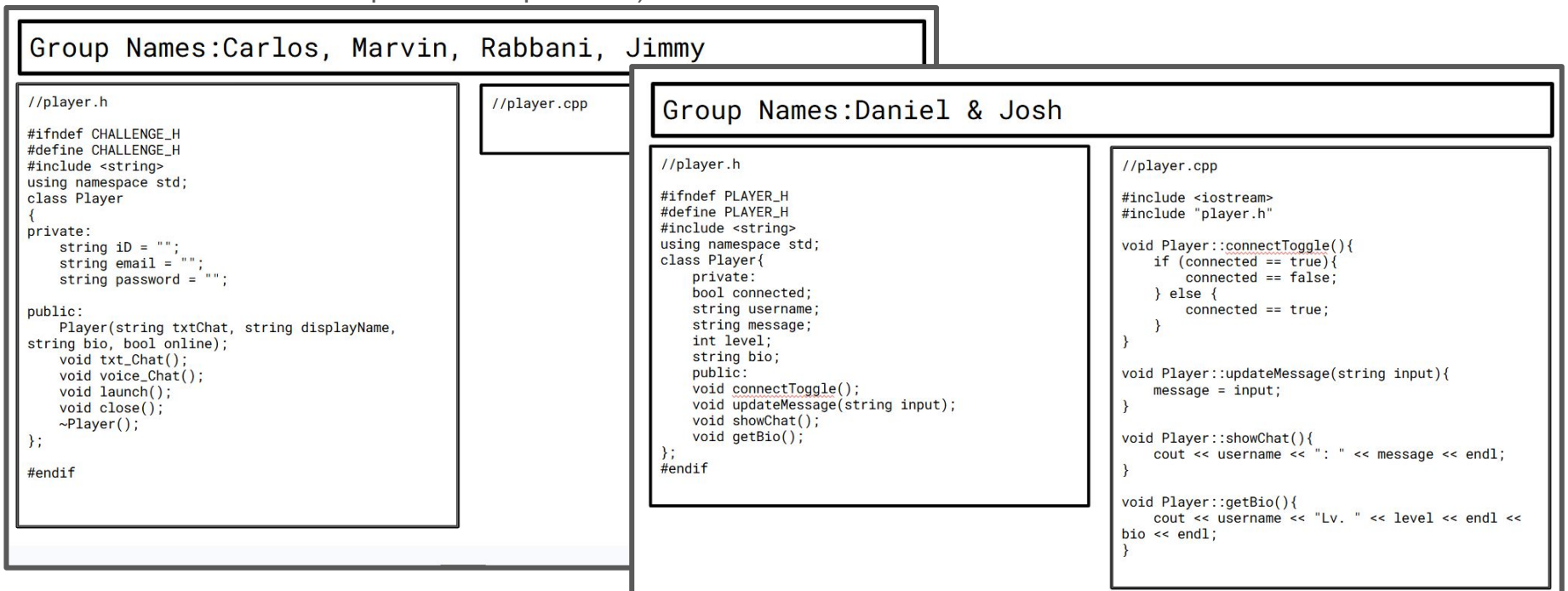
Students were given the following challenge to work on with a partner/group:



Mirrors in Object Oriented Programming

Students are encouraged to design objects and model problems in ways that are meaningful to them:

- Each students' class design becomes a mirror.
- The mirror is a reflection of their personal reasoning, creativity and interpretation of the problem (based on each student's personal experience)



Group Names:Carlos, Marvin, Rabbani, Jimmy

```
//player.h

#ifndef CHALLENGE_H
#define CHALLENGE_H
#include <string>
using namespace std;
class Player
{
private:
    string iD = "";
    string email = "";
    string password = "";

public:
    Player(string txtChat, string displayName,
string bio, bool online);
    void txt_Chat();
    void voice_Chat();
    void launch();
    void close();
    ~Player();
};

#endif
```

```
//player.cpp
```

Group Names:Daniel & Josh

```
//player.h

#ifndef PLAYER_H
#define PLAYER_H
#include <string>
using namespace std;
class Player{
    private:
    bool connected;
    string username;
    string message;
    int level;
    string bio;
    public:
    void connectToggle();
    void updateMessage(string input);
    void showChat();
    void getBio();
};
#endif
```

```
//player.cpp

#include <iostream>
#include "player.h"

void Player::connectToggle(){
    if (connected == true){
        connected == false;
    } else {
        connected == true;
    }
}

void Player::updateMessage(string input){
    message = input;
}

void Player::showChat(){
    cout << username << ": " << message << endl;
}

void Player::getBio(){
    cout << username << "Lv. " << level << endl <<
bio << endl;
}
```

Windows in Object Oriented Programming

- Students engage with solutions created by others.
  - This engagement serves as windows into different ways of thinking about object attributes, behaviors, and the abstractions.

- Seeing a variety of class designs helps the students appreciate that the same problem can be modeled in many valid ways (open windows of design variations for the same object).

OOP Projects that Foster Rehumanization

- “Model Your Community” OOP Project
  - Students design classes based on real people, roles, or systems in their lives.
    - This connects programming to identity and community.
- “Digital Storytelling with Objects”
  - Students create objects that interact in a narrative:
    - Human-centered programming → connections, empathy, imagination.
- “Ethics in Software OOP Simulation”
  - Model ethical dilemmas using interacting objects.
    - Students observe how design decisions impact fairness or privacy.

Reflections

- Providing structured partner time and a worksheet supported students' understanding of classes. Although collaboration started slowly, peer interaction strengthened over time.

- Students showed greater interest in game-based and real-world problems, which reinforced engagement and their passion for problem solving.

Why this matters

Rehumanizing STEM through OOP:

- Helps students see themselves inside STEM,
- Builds belonging and identity,
- Connects technical concepts to lived experience,
- Supports first-generation and underrepresented students,
- Creates holistic, compassionate, community-centered engineers.

Why this matters

For our teaching context, especially in CS/CE courses at Cal State universities, this approach resonates strongly with:

- First-generation student mentoring
- Focusing on real-life programming skills
- Motivating students through personal stories.

Challenges

When introducing Classes and Object-Oriented Programming (OOP) to CS1 students for the first time, instructors often feel obligated to provide explicit guidance on which attributes students should use.

- Many students struggle to grasp these concepts without strong, structured instruction, especially during their initial exposure.

Conclusion

- Rehumanizing STEM dimensions remind us that learners make sense of new concepts through prior experiences, community knowledge, and personal identity.

Conclusion

- Instructors' well intentioned description of which attributes students should use can narrow the space for students to engage creatively and draw on their own ways of understanding objects and relationships in the real world.

- At the same time students' difficulty with unguided exploration highlights the importance of instructional approaches that validate their backgrounds while gradually supporting them in developing agency within the conceptual space of OOP.