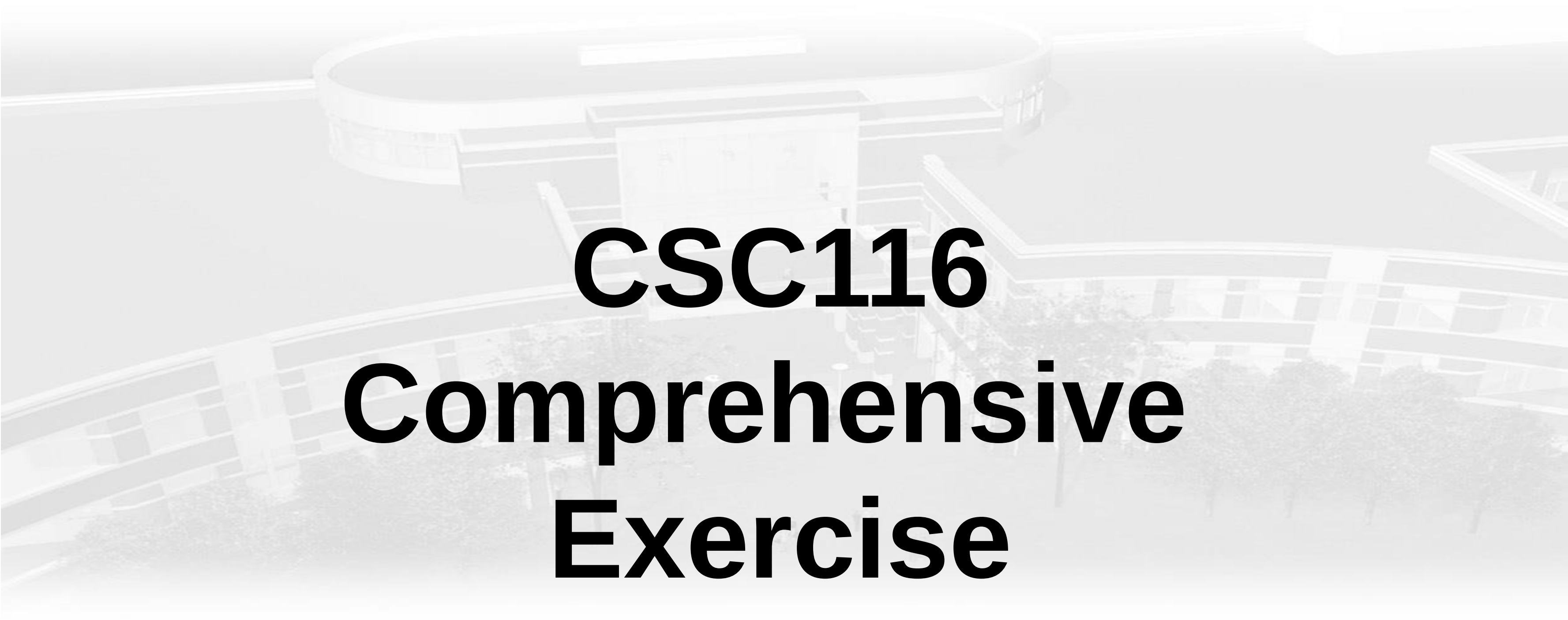

CSC116 Comprehensive Exercise

NC STATE UNIVERSITY

Department of Computer Science

Agenda

- Introduction
- Grading Information
- Project Schedule
- Complete Requirements and BBTP

Comprehensive Exercise

- Applying what you have learned in CSC116 this semester
  - Design & implement a software system from start to finish
  - Gain experience with all phases of the SDLC
  - **Work as part of a team to accomplish a collaborative goal**

CE Introduction

- You will spend the next 4 class periods working with your group to complete a project following the software development process.
  - requirements
  - design
  - implementation
  - testing
- Documentation should contain information similar to project descriptions
- Your client (your instructor) expects your work to be well planned, implemented, and tested.

Grading Information

- 2% of overall course grade
- Documentation: 85%
  - Introduction: 5% (teams) or 10% (individual)
  - Requirements: 20% (teams) or 25% (individual)
  - Design: 20%
  - Implementation: 10%
  - BBTP: 10%
  - Reflection: 10%
  - Peer Reviews: 10% (teams)
- Code - Implementation/Testing: 15%
  - Implementation: 10%
  - WB Testing: 5%

Comprehensive Exercise Day 1

- Project selection (completed)
- Requirements documentation
- BBTP
- Start Design
- Deliverable (In-class Lab 23, submit to github):
  - Project document with the following completed:
    - Introduction section
    - Requirements section
  - BBTP (no actual results)

Comprehensive Exercise Day 2

- Finish Design
- Start Implementation
- Start Testing

- Deliverables (In-class Lab 24, github):
  - Project document with the following completed:
    - Design section
    - Implementation section
  - Code to Date (and example input files if required)

Comprehensive Exercise Day 3

- Finish Implementation
- Finish Testing
- Reflection and Peer Reviews

- Deliverables (Final Project Submission):
  - – Updated Source code
  - – Updated Test program (any test data files used)
  - – Completed BBTP
  - – Final project document
  - – Peer reviews (only for those on a team)

Requirements

- The client has given you a brief, incomplete description of what is expected for the software.
- As a group, you should have a detailed discussion of *what* the software should do and *what* features it should have.
- Remember, you are not discussing design here.
- Formalize your requirements by downloading and editing the provided ComprehensiveExerciseReport.docx file. Also provide an Introduction and create a name for your project/program.

Requirements

- The following prompts are meant to aid your thought process as you complete the requirements/analysis portion of this exercise.
- After reading the client's brief (possibly incomplete description):
  - write one sentence that describes the project (expected software) and list the already known requirements.
  - What questions do you have for the client? Are there any pieces that are unclear? What user scenarios have not been clearly addressed? Compile a list of questions to ask the client at the client meeting. Make sure to document the answers.

Requirements

- Does the project cover topics you are unfamiliar with? If so, look up the topics and list your references.
- Describe the users of this software (e.g., small child, high school teacher who is taking attendance).
- Describe how each user would interact with the software
- What features must the software have? What should the users be able to do?

Blackbox Test Plan

- We are completing the black box test plan prior to the design because the black box tests should only be based on your requirements and should work independent of design.
- You currently know the program input and output based on the requirements. How exactly the user will interact with the system/program will come out of the design; you can be general with this for now

Blackbox Test Plan

- The following prompts are meant to aid your thought process as you complete the black box testing portion of this exercise.
  - What does input for the software look like (e.g., what type of data, how many pieces of data)?
  - What does output for the software look like (e.g., what type of data, how many pieces of data)?
  - What equivalence classes can the input be broken into?
  - What boundary values exist for the input?
  - Are there other cases that must be tested to test all requirements?

Blackbox Test Plan

- Download and complete the BlackBoxTestPlan.docx file provided on Moodle.
- The Actual Results column of the black box test plan will remain blank for now.
- Remember to test each equivalence class, boundary value, and requirement.