

Graduate Research Project (2-3 student teams) - 250 Points

- Topic Proposal Submission (10 points) - PDF
- Topic Outline (25 points) - PDF
- Topic Peer Review (25 points) - PDF
- First Draft (40 points) – Microsoft PowerPoint
- Pecha Kucha (50 points + Extra Credit) – Microsoft PowerPoint (with timing)
- Final Presentation (100 points + Extra Credit) – Microsoft PowerPoint

Introduction

The goal of this work is to perform scholarly research on a topic related to object-oriented concepts, languages, frameworks, testing, principles, libraries, or patterns, or other uses of patterns in computer science (e.g. web, architecture, anti-patterns, etc.). Any type of research project is acceptable, including: analytic, argumentative, defining, compare/contrast, interpretive, experimental, or survey. The topic may not be proprietary in any way, it must be able to be shared with reviewers and the class.

You may also use this project to look at OO elements of your own research topics, however, you may not duplicate submission of work for this project with other submissions for other classes or research requirements.

This semester the research will be presented in the form of a research presentation (40 slides or more), optionally with accompanying software examples. Please note it is expected to see incrementally more thorough work from a three person team than from a two person team for these investigations.

The first three project submissions will be via PDF. The final three will use Microsoft PowerPoint. The Pecha Kucha presentation will be used in a classroom presentation and must be provided using timed Microsoft PowerPoint.

Late Submissions

In all cases (unless restated), the submissions will follow these late penalties. A four hour window after the due date/time will be allowed without penalty. Submissions in the next 44 hours will suffer a 5% late penalty. All late submissions in the next 48 hours will be penalized 15% of the final grade assigned to the submission. No materials for the submissions will be accepted after this period (except for health/medical issues).

Topic Proposal Submission (10 points) – Due by noon, Wed 9/16.

This PDF should be one page, and should include the following:

- Topic title being proposed
- Team members
- A paragraph describing the topic content, including relation to OO and major elements
- A paragraph describing the reason for selection

You are also directed to put your topic title/subject in a shared Google Doc to avoid duplication of topics with other teams. The Google Doc is here:

<https://docs.google.com/document/d/1vvfmoUyUbO2ykPgrM-26R6IAjkmro93xwi9p8cA7JXw/edit?usp=sharing>

Once the submitted topic is approved, you will move on to the Topic Outline.

Topic Outline – Due week 6

This will be a more detailed version of the topic proposal, approximately two to three pages, including at least 5 supporting literature citations. Details will be provided when the assignment is made.

Topic Peer Review – Due week 8

Your team will review another team's graduate presentation Topic Outline. You will be required to provide at least five suggestions related to the topic, along with literature or web citations for each. Further details will be provided when the assignment is made.

First Draft – Due week 12

This is your initial draft of the research presentation. The target length for the research project presentation is 40 slides or more. **This initial draft should be over 50% of the final draft project content with appropriate citations.** It should include the team names and the topic title as well as sections similar to the initially proposed outline, although changes are allowed. If code examples are part of the project, there should be evidence of progress on that code at this point as well.

As in the final draft, this initial draft must include at least ten citations, five of which should be drawn from scholarly literature: papers, books, or similar sources. The remaining citations can come from the web. Wikipedia should not be a direct citation reference, although using it may guide you to original sources. Further details will be provided when the assignment is made.

Pecha Kucha – Due week 14

This presentation is intended to briefly provide the class and class-staff with an overview of your work on the project. Plan to include the project title, an introduction to the team members, an outline of your project (probably across several slides), and any interesting elements you think the class should know.

Your Pecha Kucha presentation should be provided using PowerPoint and should be prepared to automatically advance through 20 slides at 20 seconds per slide. The total time for the presentation will be 6 minutes and 40 seconds. Each slide should include any notes you intend to use to speak to the slide during presentation. Remember, at 20 seconds per slide, there must be very minimal text – consider use of graphics and images that you will talk to wherever possible. This may also make it difficult to directly use slides from your more detailed research presentation. There will be a 10% point penalty will be assessed if the presentation does NOT automatically advance at 20 seconds per slide.

A guideline to creating Pecha Kucha presentations can be found here:

<https://wiki.rice.edu/confluence/display/DMCGUIDES/Creating+a+Pecha+Kucha+Presentation+Using+PowerPoint>. You should plan on your team presenting the Pecha Kucha for your topic in class (although some presentations may not be scheduled due to time available), but you must submit the Pecha Kucha on or before the due date for credit regardless.

Final Presentation – Due week 16

Your final project submission with appropriate citations will be submitted as a 40 slide or more PowerPoint presentation, along with any code examples. There will be 5, 10, and 20 extra credit point quality awards made for the best final projects submitted, as judged by the instructor.

Typical Past Topics

- Java: Is its dominance in industry numbered or just sailing through rough waters?
- TDD in Python
- Generating code solutions from UML
- Comparing the practical applications and performance of Java vs C++
- Keras vs PyTorch: An Object Oriented Comparison
- Exploring Kafka Framework using Object Oriented Paradigm
- Analysis of Spring framework
- Hibernate framework & data persistence - an analysis
- A Modern Survey of Multiple Inheritance in Object-Oriented Languages
- UML in object-oriented analysis and design
- Node.js in Object Oriented Analysis & Design
- Python in Data Science-Multi-processing vs Threading
- Java vs Node.js comparison
- An OOP Approach to ROS Development
- Automated selection of data structures and algorithms
- OO Natural Language Processing Frameworks
- The Benefits of an Object-Oriented Tool for Interdisciplinary fNIRS Analysis
- Examining an AR-Facilitated Human-Robot Negotiation Study as an Object-Oriented System
- Effect of object-oriented measures on external quality attributes of a system
- Object oriented principles in Scala Vs Java
- The Evolution of Object-Oriented Programming: Comparing its Origins with Modern Day Practices

Note: It's common to do comparison projects, but I expect such comparisons to be thorough and thoughtful, and if comparing languages or libraries, to include code examples.