📜 265 Course Syllabus (Draft from Spring 2021)

> ✏️ **The syllabus is a crucial part of understanding expectations and responsibilities for the course. It is EVERY STUDENT'S responsibility to know, understand, and agree to the terms of the syllabus. Please read through this page carefully.**
>
> **Don't forget to complete the <u>action items</u> to-do list at the end of the document!**

ITP 265 Object Oriented Programming

Instructor: 👩 <u>Kendra Walther</u>

Office: RRB 219 (but on Zoom for Spring 2021)

Office hours: (See ☎️ <u>Full Course Staff Contact Info</u>)

Email: <u>kwalther@usc.edu</u>

Course Sections: ☕ 10:00-11:50am, Tu/Th (Section 31848)

☕ 2:00-3:50pm, Tu/Th (Section 31849)

Contact Info: All general course/assignment questions should be asked on Slack or Piazza (every student will receive an invitation at the start of the semester

🔑 Enrollment

Prerequisite(s): ITP 115 or ITP 165

Units: 4 → According to <u>USC Policy</u>, this means that you are expected to work 8 hours per week outside of class time.

Concurrent Enrollment: course(s) that must be taken simultaneously

Format: This course will make use of several tools for content and assignments including Google Drive, Blackboard, and an online class discussion forum. Lecture notes and any supplemental course content will be posted to Notion and/or Google Drive and/or Blackboard for use by all students. Any and all announcements for the course will be posted to the class discussion forum. All assignments will be posted to Blackboard and will be submitted through Blackboard.

📜 Course Description

This course focuses on problem solving within the object-oriented programming paradigm. This is the second course in the introductory series for the programming minor. Students will expand upon what they learned in their introductory programming course, applying it to the **Java** programming language. Students will learn how to design and create classes in Java using constructors, accessors, and mutators to maintain object state. The course focuses on object-oriented programming design, and students will learn about inheritance, polymorphism, abstract classes, and interfaces. Students will learn best practice approaches for software project design using object-oriented principles and some basic design patterns. Students will be introduced to collection classes and how to use basic data structures. By the end of the course, students should feel comfortable designing a system with multiple classes using inheritance.

Course Learning Objectives

- Build and strengthen programming and software design skills
- Understand the difference between classes and objects
- Design classes within Java to represent real-world data
- Understand basic object-oriented principles such as inheritance and polymorphism
- Use the Java collection classes to solve real-world problems

- Design a system with multiple classes using inheritance

💼 Course Notes and Tools

Kendra will post lecture notes, videos, code, and other supplemental course content for use by all students enrolled in the course. **Students may not share the material outside of the course or post to any online location.**

Announcements for the course will be posted on one of the platform tools and sent to your USC email address. Synchronous Participation Activities will be submitted on Slido. All other graded course assignments will be be submitted through Blackboard unless otherwise noted.

Course Structure

This class meets for two hours twice a week. Class time will comprise of lecture and various in class individual or group activities. I will be giving several quizzes to assess learning in lieu of a larger midterm and final. Programming assignments and the final project will be assigned to be completed outside of class time. Access to a functional computer where you can install software is required. ITP has a laptop loaner policy for students enrolled who do not have a personal laptop.

All class meetings will be set up to support synchronous and asynchronous learners. Each meeting will have both in-class participation activities and a corresponding Blackboard discussion forum for asynchronous participants to complete activities. Synchronous participants will be given opportunities during the live class meeting to complete activities (either using Slido or on Blackboard). Asynchronous participants will have 60 hours from conclusion of class lecture to watch the recorded video and then post to the corresponding Blackboard discussion forum summaries or responses to any exercises done in class.

Course Tools

This course will make use of several tools for content and assignments, including Notion, GoogleDrive, Blackboard, Slido, Slack, and Piazza. **It is the student's responsibility to understand how and where class information is located, or to ask for help if something is unclear.**

In general, this Notion site will contain the most up-to-date course information, so I highly recommend downloading the Notion app to your device(s).

Communication Outside of Class

I encourage you to ask questions and get help. We offer lots of office hours and an online forum for asking general questions (of the whole class or privately to the instructional team).

Zoom Participation

All lectures and office hours will occur via Zoom, a high-quality, mobile-optimized, video and audioconferencing service free to all USC students.

- Zoom will allow us the opportunity to **engage face-to-face** – even from a distance.
- Though not required, I **encourage** you all to attend lecture with **video capability on.**
- All Zoom **lectures will be recorded** and shared with the class each week.
- If you do not have a USC Zoom account yet, activate and configure your USC Zoom account using the instructions found on the Keep Teaching

It is expected that students on Zoom behave professionally, treat others with courtesy and respect, use language thoughtfully, wear appropriate clothing, and avoid inappropriate surroundings or inappropriate or distracting virtual backgrounds. Ideally you will join class from a suitable, quiet location, with a device that permits full participation in the class activities.

I do not like teaching to a bunch of named boxes, so I encourage students to turn on video, **IF** you are comfortable doing so (it also helps everyone feel a bit more connected). However, I understand that not everyone may be comfortable or able to share their video at all times, so I expect students to set up Zoom to display a photograph or bitmoji image when the video is off.

Slack Channel

To supplement and enhance in-person classroom interactions, we will also use Slack, a collaboration hub where the right people are always in the loop and key information is always at your fingertips.

- Using Slack, I will **share information** for our course throughout the semester.
- As a student in this course, you will be able to use our channel to collaborate with your peers, teaching

All general course/assignments questions should be asked on Piazza or Slack. Every student will receive an invitation at the start of the semester. Students who add late may need to reach out to the instructor in order to be manually added to the platforms. For other questions or concerns, please email: kwalther@usc.edu General timeline for replying to emails is within 24 **business** hours, but turn around time is often much quicker.

assistants, and myself in **Q&A, office hours,** or **project channels**.

- Prior to the start of classes, please be sure to **set up your Slack account** at https://usc.enterprise.slack.com/ and familiarize yourself with our class channel. Additional reference guides and resources can be found on the Keep Teaching

- The expectation will be that you will **use Slack** as tool **to be successful in the course.**

- As a reminder, all **Slack conduct must be in line with** USC policies and USC Policies Student Issues. Any behavior that breaks outlined policies will be subject to discipline.

🏆 Grading

Breakdown

Book activities are recommended for most students. I used to have students choose one of the following options for grading. This semester I will calculate your grade both ways and give you the higher of the two. **We will discuss the rational behind letting students opt into the book or not during the first week of class.**

Option 1: Completing online textbook activities

- Assignments (weighted proportionally)   → 35%
- Reading Activities (online book)   → 5%
- Participation (Activities, labs, forums)   → 15%
- Final Project (due end of semester)   → 20%
- Quizzes   → 25%

- **Total**   **→ 100%**

Option 2: No Textbook activities

- Assignments (weighted proportionally)   → 35%
- Participation (Activities, labs, forums)   → 15%
- Final Project (due end of semester)   → 20%
- Quizzes   → 30%

- **Total**   **→ 100%**

Grading Scale

A >= 94

A- >= 90 and < 94

B+ >= 87 and < 90

B >= 83 and < 87

B- >= 80 and < 83

C >= 73 and < 77

C+ >= 77 and < 80

C- >= 70 and < 73

D+ >= 67 and < 70

D >= 63 and < 67

D- >=60 and < 63

F < 60

If you are taking the class with a grade of **P/NP,** you must earn a grade of **70%** or higher in order to receive a P.

Reflection Journals on Blackboard... I have created a space each week (or two) for you to share personal reflections about your learning experience. These are not required for your grade, but I think reflection is an important part of learning, so I will offer a small incentive ("round up") for students who complete the majority of these.

📚 Readings

The textbook for the course is recommended for most students. Most previous students had a very positive experience with the online book and recommended it for use future semesters. However, a small subset of confident students with more programming experience found the book redundant, and so I offer students two grading options, and allow you to choose to include the book activities or not.

The online book for this course will be customized version of Java Early Objects with zyLabs by Roman Lysecky and Adrian Lizarrago through an online textbook platform, ZyBooks: https://www.zybooks.com/catalog/java-early-objects/   A book subscription costs $58. The book includes participation activities and challenge exercises. Participation activities in each section must be completed in order to receive reading credit. All challenge exercises will be extra credit (in the book activity portion of the grade) and counted at the time of the corresponding assessment for the chapters.

Zybooks Reading: Recommended for most students. See 📖 Chapter 1 for how participation and challenge activities are graded.

1. Sign in or create an account at learn.zybooks.com **USE YOUR USC EMAIL ADDRESS**
2. Enter zyBook code: USCITP265WaltherSpring2021
3. Subscribe

🥏 Course Policies (read thoroughly)

Course Material Policy

Do not reproduce, distribute, or post any lecture material, assignments, quizzes, or exams publicly without my explicit written consent. Students may take notes and make copies of course materials for their own use. Students may **not** post any ITP 265 course materials on any online sites. Doing so is a copyright violation and an academic integrity violation that will be reported and dealt with accordingly.

Participation and Attendance

Attendance is not mandatory, but students are responsible for any in class work or participation activities, announcements made during lecture time, and for understanding material covered in class. All student work will be graded on the assumption that they have mastered material from class. The instructor will give verbal instructions for completing the in-class participation activity, students who miss the in-class participation activity will need to complete the asynchronous Blackboard activity for the class period. It is the student's responsibility to attend class or watch the entirety of the recorded class Zoom session.

It is entirely the student's responsibility to complete any in-class participation work or the corresponding asynchronous assignment. In order to keep up with the class material, the weekly participation work must be completed during lecture time **OR** within 60 hours of the conclusion of the course lecture time for asynchronous participants.

DSP Accommodations

If you have course accommodations authorized by DSP (Disability Services and Programs), please email the instructor your accommodation letter **by the end of Week 3**, the subject of the email should be "*ITP 265 Course Accommodations*". In the body include your name and your

Programming Assignment Policies

Programming assignments will generally be due one week after they are assigned and should be completed individually. However, some assignments take longer than a week, so I do not have a standard due date and time for assignments in this course.

All code should be submitted on Blackboard (unless otherwise stated) and must compile. It is the student's responsibility to double-check that Blackboard submission finished uploading properly and that the **correct** files were uploaded. (Double check the return receipt email) **Assignments that are resubmitted for grading due to user error will be deducted points for being late.**

Backups

Students should keep a copy of all of their assignments. Frequent backups to an external drive or to the cloud is strongly recommended. ITP is not responsible for any work lost, nor will we give extensions for losing work due to failure to backup work

Grading Timeline and Regrades

Assignments will be graded and students will receive feedback within two weeks after submission.

Students will have two weeks after graded feedback is given to contest scores (e.g. assignments, tests, exam, and project). After two weeks, scores will not be changed. To contest a grade, create a private post in the grading questions folder on the discussion forum with your name, section, assignment name, and your reasons for requesting the regrade. This will allow the grader **and** instructor to view your submission and make a decision. Please do not directly email the grader without also CC'ing the instructor.

Late Policy

class section (*Coffee or NightOwl*). In addition, reach out the week before each quiz to discuss details for coordinating specific exam accommodations.

Late Add

Per university policy, students are allowed to add the course after the initial start period. Any students wishing to add the course should plan on attending the course from the beginning of the semester. Upon adding the course after week 1, the student should email the instructor **immediately** to make sure there is a plan for completion of work and learning missed materials. Any missed work is required to be completed and submitted according to the schedule provided by the instructor, in general within a week of the course add date.

Quiz Policy

No make-up exams, quizzes, or tests (except for **documented** medical or family emergencies) will be offered. If a medical or family emergency occurs, it is your responsibility to provide adequate documentation as soon as possible to the instructor.

Etiquette and Electronic usage

Additional Zoom etiquette norms are posted above. Participation during class is crucial, so students should engage with the instructor by using the Zoom chat, raising their hand, and/or using slido, a platform that allows for polling and anonymous questions.

It is the student's responsibility to submit assignments **on or before** the due date. We will give a 10 minute grace period for connectivity issues. Assignments turned in up to 12 hours late will have **10%** of the total points deducted from the graded score. Assignments turned in 12-36 hours late will have **30%** of the total points deducted from the graded score. Assignments turned in 36-72 hours will have **50%** of the total points deducted from the graded score. After three days, submissions will **not** be accepted, and will result in a score of **0** (zero).

Free late

Each student will be allowed **THREE** 24-hour late assignments for "free", which may not be used on final project submission. These may not be broken down until smaller time units, but can be used together for one 76 hour extension. Students must indicate that they are using a Free Late in the comment field of Blackboard **AND** fill out the "Late Assignment" Google form.

Support Systems

USC offers a variety of student-focused support, please see this document for the full list with contact information:

🧑 Copy of USC Statement on Support Systems and Academic Conduct

Academic Integrity

USC seeks to maintain an optimal learning environment. General principles of academic honesty include the concept of respect for the intellectual property of others, the expectation that individual work will be submitted unless otherwise allowed by an instructor, and the obligations both to protect one's own academic work from misuse by others as well as to avoid using another's work as one's own. All students are expected to understand and abide by these principles.

😢 Plagiarism

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in **SCampus** in Part B, Section 11, "Behavior Violating University Standards" https://policy.usc.edu/scampus-part-b/ . Other forms of academic dishonesty are equally unacceptable. See additional information in **SCampus** and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct.

Examples of behavior violating University standards:

- The submission of material authored by another person but represented as the student's own work, whether that material is paraphrased or copied in verbatim or near-verbatim form.

- Obtaining for oneself or providing for another person a solution to homework, a project or other assignments, or a copy of an exam or exam key without the knowledge and expressed consent of the instructor.

- Unauthorized collaboration on a project, homework, or other assignment.

If the instructor, a grader, or a lab assistant **suspects** you of academic dishonesty, it has to be reported to SJACS (https://sjacs.usc.edu). Do not share assignments with any other people. Do not submit another person's work as your own. Do not look at other students' papers during exams. Do not leave the room during an exam without permission. **Do not cheat! As Trojans, we are faithful, scholarly, skillful, courageous, and ambitious.**

- Fabrication: Submitting material for lab assignments, class projects, or other assignments which is wholly or partially falsified, invented, or otherwise does not represent work accomplished or undertaken by the student.

Academic integrity tutorials can be found at https://libraries.usc.edu/research/reference-tutorials

🤝 Viterbi Honor Code

Engineering enables and empowers our ambitions and is integral to our identities. In the Viterbi community, accountability is reflected in all our endeavors.

Engineering+ Integrity.

Engineering+ Responsibility.

Engineering+ Community.

Think good. Do better. Be great.

These are the pillars we stand upon as we address the challenges of society and enrich lives.

📅 Course Schedule

The 📅 Weekly Overviews on Notion contains details about each week of class, as well as containing links to homework and reading assignments. Other course materials can be found (in a variety of views) on the 📄 All Course Materials page. Each named item (weeks, assignments, reading) is also a link to a page that may be opened to find further details about course activities. Everything that you need for the course should be found on (or linked from) the 📅 Weekly Overviews except for the homework submission links on Blackboard.

Weekly Overviews

Aa Name	≡ Overview	≡ Learning Objectives	📅 Date Range	≡ Assignment	≡ Related Content (Other Readings or Videos)	≡ Zybooks	≡ Le Co Ex
Study Days	Special Office Hour Schedule		@May 1, 2021 → May 4, 2021				
Exam Period	Final Exams		@May 5, 2021 → May 12, 2021	Final Project Due Saturday May 8 by midnight			
Weekly Overview							

Aa Name	Overview	Learning Objectives	Date Range	Assignment	Related Content (Other Readings or Videos)	Zybooks	Le Co Ex
🛠 Week 01	`No class Mon/Tues` Overview of Course. Syllabus questions. Demo of Zybooks. Java Program Anatomy. IDE Setup.	Setup and become comfortable with IDE. Perform basic coding constructs using Java. Build a supportive class community.	@January 18, 2021 → January 23, 2021	💻 Install BlueJ	📜 Java Syntax Basics	📖 Chapter 1: Programming Basics	
⭐ Week 02	Primitive Data. Conditionals. Looping. Strings. Input with `Scanner`. OOP Overview. Classes and Objects.	Gain comfort using basic programming concepts (conditionals, loops, and methods) in Java. - Explain how the Java API is set up and how to interpret the text to use the basic classes like `Math`, `Random`, and `String`. - Experiment with using `Scanner`'s methods in code to get user input. - Explain problems with `Scanner` for console input and how to overcome problems. - Describe and give examples for OOP terms (class, instance, object, methods).	@January 25, 2021 → January 30, 2021	💻 HW00: ASCII Animation due Thursday Jan 28	📜 Binary Numbers (Base 2) 📜 Strings and Scanner	📖 Chapter 2: Java Basics	

Aa Name	☰ Overview	☰ Learning Objectives	🗓 Date Range	☰ Assignment	☰ Related Content (Other Readings or Videos)	☰ Zybooks	☰ Le C Ex
🧑‍🎓 Week 03	Creating classes and objects. Constructors, Accessors, Mutators. CyberPet Activity. Create `UI` class.	Design and code classes in Java to represent real-world data. - Explain why accessor and mutator methods are needed and demonstrate ability to code methods following template models. - Explain purpose of constructors and write a constructor using instance variables. - Generate additional constructors without repeating multiple lines of code. - Understand and use `this` reference in code. - Write a `toString` method for a class - Write a simplified `equals` method for a class	@February 1, 2021 → February 6, 2021	📇 HW01 Health Records due Thursday Feb 4	🔑 Object Oriented Programming Key Terminology	📖 Chapter 3: Classes and Objects	

Aa Name	☰ Overview	☰ Learning Objectives	📅 Date Range	☰ Assignment	☰ Related Content (Other Readings or Videos)	☰ Zybooks	☰ Le Co Ex
💰 Week 05	No class Mon/Tues Enumerations. Arrays.	Successfully use arrays when coding in Java. (Including arrays of primitives, arrays of objects, perfect-sized arrays, and oversized or partially filled arrays) - Use Enums in real-world coding situations. - Apply OOP to real-world coding solutions.	@February 15, 2021 → February 20, 2021	💳 HW03: CreditCard (Code with a Friend) Due Saturday Feb 20	🖨 OOP Basics (A Video Review) 📒 OOP FAQs 🖨 Week05: Enum Lecture	📖 Chapter 4: Arrays	
💳 Week 04	Class Design. Intro to UML class diagrams. Array Syntax. Using 1d arrays.	Interpret basic UML class diagrams and use them to guide writing a new class in Java. - Given a problem statement, design a UML class diagram and write code based on the diagram. - Successfully use arrays when coding in Java. (Including arrays of primitives, arrays of objects, perfect-sized arrays, and oversized or partially filled arrays)	@February 8, 2021 → February 13, 2021	💳 HW02: BookTeaque due Saturday Feb 13	🖨 OOP Basics (A Video Review) 📒 OOP FAQs		

Aa Name	≣ Overview	≣ Learning Objectives	🗓 Date Range	≣ Assignment	≣ Related Content (Other Readings or Videos)	≣ Zybooks	≣ Le Co Ex
🧁 Week 06	Enums and Arrays. Intro to Lists (`ArrayList)`	Explain difference between arrays and ArrayList Define encapsulation Explain at least one difference between ArrayList and LinkedList Initialize ArrayList in code using generics syntax Recognize ArrayList methods	@February 22, 2021 → February 27, 2021	HW04 Simple Arrays due Thursday Feb 25		📖 Chapter 5: Arraylists	
🍪 Week 07	ArrayList JOptionPane 2d arrays Eclipse	Use ArrayList objects and methods in real-world coding situations. Gain familiarity with JOptionPane Use JOptionPane to: show the user message, ask the user yes/no questions, get String input from the user, and get input based on an array of options using a drop-down menu or buttons of options. Use 2d arrays in a program Become familiar with features of Eclipse IDE	@March 1, 2021 → March 6, 2021	HW05: African American Achievements (list homework) due Saturday March 6			

Name	Overview	Learning Objectives	Date Range	Assignment	Related Content (Other Readings or Videos)	Zybooks	Le... Co... Ex...
🌻 Week 08	Using inheritance. Polymorphism of objects. `Object` Class + `equals` method	Become familiar with features of Eclipse IDE Use JOptionPane to: show the user message, ask the user yes/no questions, get String input from the user, and get input based on an array of options using a drop-down menu or buttons of options. Define inheritance Use Inheritance to define subclasses.	@March 8, 2021 → March 13, 2021	HW06: Lights Out Game (2d array) due Saturday March 13	🌙 Eclipse Help	📖 Chapter 6: Inheritance	
🧑‍🤝‍🧑 Week 09	`Object` methods. Abstract classes. Interfaces. `Comparable` interface and compareTo method Polymorphism	Demonstrate polymorphism in a program with inheritance. Describe the difference between overloading and overriding. Define abstract classes and interfaces. Demonstrate polymorphism in a program with interfaces.	@March 15, 2021 → March 20, 2021	Mid Semester Course Feedback Form (anonymous)		📖 Chapter 7: Abstract and Interfaces	

	Name	Overview	Learning Objectives	Date Range	Assignment	Related Content (Other Readings or Videos)	Zybooks	Le... Co... Ex...
	🏡 Week 10	No class Mon/Tues Program Design, using Polymorphism.	Understand the purpose of equals and comparable in programs. Use inheritance in Program Design.	@March 22, 2021 → March 27, 2021	HW07: Product Inheritance Part 1 due Saturday March 27	📽 Week 10, Day 2 Video Lecture (Hour 1 of class)	📖 Chapter 8 Testing (Super Short Chapter)	
	🗺 Week 11	Maps	- Understand the structure of a map (dictionary, key:value) data structure - Design programs using the map data structure Use a HashMap to efficiently store data in a program. Be fluent with hashmap API and using methods to access data in a program. Explain and define the difference between lists, maps, and arrays. Choose the best data structure for a problem and construct code using that structure.	@March 29, 2021 → April 3, 2021	HW07: Product Inheritance Part 2 due Thursday April 1	Read 📜 FAQ: Polymorphism, Inheritance, Interfaces, and Abstract Classes 📜 FAQ Maps	📖 Chapter 9: Maps and Files	

	Name	Overview	Learning Objectives	Date Range	Assignment	Related Content (Other Readings or Videos)	Zybooks	Le... Co... Ex...
💾 Week 12	`No class Wed/Thu` Reading and Writing to Files Catching Exceptions Exception Handling.	Classify exceptions that can be dealt with by better programming versus ones that are handled in code. Write a program that reads and writes data to/from a file.	@April 5, 2021 → April 10, 2021	HW08: Maps due Saturday April 10		📖 Chapter 10: Exceptions	
🦇 Week 13	Java Collection Framework. searching and sorting. OOP Design Principles	Describe the structure and purpose of the Java Collection Framework Understand the importance of making flexible software. Use proven OOP principles to make well-designed software.	@April 12, 2021 → April 17, 2021	HW09: Files due Saturday April 17		📖 Optional Chapter 11: Collections and Generics	
💃 Week 14	`No class Wed/Thu` OOP Design Patterns	Match design pattern names with their descriptions. Give examples of how design patterns facilitate large programming projects. Use one or more design patterns in a program.	@April 19, 2021 → April 24, 2021	HW 10: FInal Project Plan due in class April 26 or 27		📖 Chapter 12 and 13 (Optional)	

Aa Name	≡ Overview	≡ Learning Objectives	📅 Date Range	≡ Assignment	≡ Related Content (Other Readings or Videos)	≡ Zybooks	≡ Le Co Ex
🏘️ Week 15	OOP Design Patterns Final Projects Miscellaneous Topics	Design (and program) a project using the MVC paradigm to organize code. Use UML to model the design for a complex software system.	@April 26, 2021 → April 30, 2021	🎉 Final Project Overview			

🧠 Final Project

Requirements

The initial **design** for final project will be assigned and graded as the last course homework (Homework 10 or 11). Students should immediately start programming their final projects and will submit **two** intermediate graded checkpoints. The final project will be due by midnight on Saturday May 8th.

Students must plan and implement a multiple-class, fully functioning application in Java. Successful projects will have a clear inheritance hierarchy, read and store data to files, allow for user interaction, and demonstrate concepts learned during the course (like inheritance, polymorphism, interfaces, MVC pattern, and good code style).. A project must represent the student's sole effort; online tutorials or other examples may be consulted, but they must be improved upon and noted in the final documentation. Failure to note and provide links to reference material will be considered cheating. The final project will be graded on how it fulfills the requirements and the quality / completion of the code.

Final Project Grading.

Each of the below categories will be rated on "Approaching Mastery", "Attemping Mastery", and "Incomplete or Not Functional"

- Final Project Intermediate Work (checkpoint 1): 10 points.
- Final Project Intermediate Work (checkpoint 2): 10 points.
- Peer feedback: 5 points
- Inheritance Hierarchy and code implementation of hierarchy: 20 points
- Data to files: 20 points
- User Interaction: 15 points
- System representation and use of collections: 10 points
- Coding Style: 5 points
- Final Report: 10 points
- Reflection: 10 points

Total points possible: 115

Closing Statements

We're in this together! **

To begin, these are unprecedented times for all of us. I know that many of you are dealing with a lot of anxiety and uncertainty. This is a new experience and we're going to do what we can to make it work. I taught online this summer and learned a lot of lessons doing so; but the most

Principles for Learning during a Pandemic *

1. Nobody signed up for this.
   - Not for the sickness, not for the social distancing, not for the sudden end of our collective lives together on campus.

important thing that I learned is that we need to be kind and flexible.... Kind to ourselves and each other, and flexible as we attempt to learn while 2020 and 2021 and all the stressors that affect our lives (*including but not limited to this pandemic, hurricanes and power outages, political unrest, racial equity, international student stress, financial hardship, and whatever family stress may occur as we all try to live and work under one roof.*)

I know that everyone does not have the same access to resources, your time is being spent differently, and your environment is possibly vastly different than the "ideal" learning environment of a face-to-face classroom. I will be up front with you all that I am teaching this class from a corner of my bedroom while also parenting/over-seeing the at-home learning schedule of my two kids. It is not the ideal teaching environment, but I have been (and will continue to) do my best to make it work. Although this is not our ideal version of the spring semester, I am confident that we can work together to make the most of our experience and support each other through the semester and meet the course learning goals, even if we have to adjust our expectations to do so. I promise to work hard to continue to redesign course elements to support you and your learning in this online environment.

If you're experiencing any problems related to your ability to participate in this course, please let me know ASAP. I will be as flexible as I can be and adjust to the situation, but I cannot do anything to help if I am not informed. For each class there will be options for synchronous and asynchronous participants, I will record sessions and provide clear instructions on what to do; office hours will be scheduled at different times of the day to accommodate learners in different time zones, and I will have flexible 1:1 appointment times available as well. I am still working on making sure all the apps that I use to supplement the course are available to everyone and USC still has not finished setting up some of the tools that I expect to use - so please be patient as many things may need adjusting depending on when they are rolled out and based on individual student needs.

*** description adopted from a version shared by Kate Pierce at UW.*

- Not for an online class, not for teaching remotely, not for learning from home, not for mastering new technologies, not for varied access to learning materials.

2. The humane option is the best option.
   - We are going to prioritize supporting each other as humans.
   - We are going to prioritize simple solutions that make sense for the most.
   - We are going to prioritize sharing resources and communicating clearly.

3. We cannot just do the same thing online.
   - Some assignments are no longer possible.
   - Some expectations are no longer reasonable.
   - Some objectives are no longer valuable.

4. We will foster intellectual nourishment, social connection, and personal accommodation.
   - Accessible asynchronous content for diverse access, time zones, and contexts.

5. We will remain flexible and adjust to the situation.
   - Nobody knows where this is going and what we'll need to adapt.
   - Everybody needs support and understanding in this unprecedented moment.

* From Brandon Bayne, UNC Chapel Hill

Syllabus Action Items

☐ (*If applicable*) DSP accommodation letters need to be sent to instructor

☐ **Zoom**: Upload your photograph or bitmoji image for display when your video is off. (Sign in to the website usc.zoom.us and click **Profile** in order to add or change your image.)

☐ Post any syllabus questions or clarifications needed on the Audience Q&A section of Slido (or go to slido.com and use eventCode #itp265) - or upvote questions that are already there!