CIT-244-Z01 – Object-Oriented Design and Data Structures Using Java

COURSE OUTLINE

Class Section(s):	**CIT-244-Z01**	Class Location:	Online via blackboard
Class Time(s):	Online via blackboard	01 / 30 – 05 / 07	

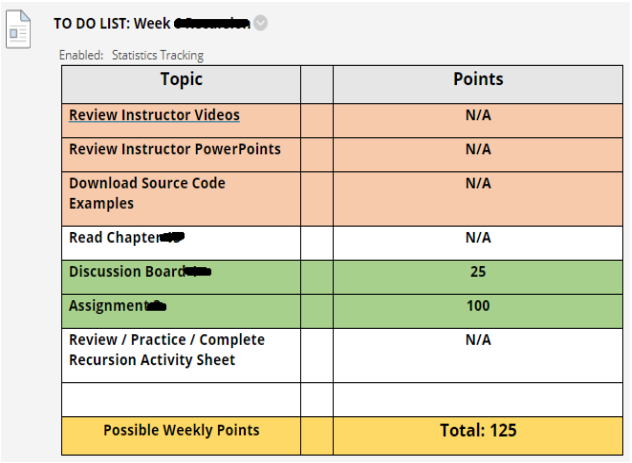
Instructor:	**James Winyard**	Semester:	**Spring 23**
Office Hours:	M: 12:15 PM – 2:00 PM T: 12:15 PM – 1:00 PM W: 12:15 PM – 1:45 PM H: 09:00 AM – 10:00 AM	Office Location:	M342 (Allegheny Campus) Zoom: https://ccac.zoom.us/j/93566978994?pwd=dTQ2UDFwR29YTWd6MEdnSThNekliZz09 ***Note:** This semester I will be holding all office hours via Zoom **but** I will also be in my physical office location M342 (Allegheny Campus) during these times, if you prefer to stop by physically.*
Instructor Contact Methods:	Email *(Preferred):* jwinyard@ccac.edu Office Phone: 412-237-6526 Office Hours Link: During scheduled hours click here		

Course Credits:	4
Prerequisites / Co-requisites:	CIT-130
Course Description:	This course builds on the concepts of software design and development introduced in CIT 130. The focus of the course is on Object-Oriented Programming (OOP) design, advanced Graphical User Interface (GUI), algorithm analysis and efficiencies, recursion, data structures and predefined Application Programming Interfaces (API). Additional course topics include file input/output processes and multi-threading. Java 8 will be used, with the NetBeans IDE.

Books & Materials	**Book** Starting Out with JAVA from Control Structures through Data Structures 4th Edition **ISBN: 978-0-13-478796-1** **Materials** • A folder to organize handout examples. • 2GB (minimum) thumb drive to store class files. • Internet access to download/submit assignments to and from blackboard site. • Java JDK *(8 Preferred)* • NetBeans IDE *(8.2 Preferred)*
Open Lab, Tutoring, etc.	Tutors are located in the Learning Assistance Center (LAC, L309), which is in the Library Building. An appointment is needed. **(NOW VIA ZOOM)**

Email Etiquette:
I am happy to answer emails with any questions or concerns you may have involving anything from our course. With this said, when writing your **email please include the Course number, Section, and Assignment/Topic | Example: CIT-244 Z01 Assignment 2 Question** with several classes and various sections for each, this will allow me to narrow down exactly what you need help with, without having to dig through a class roster to ensure a prompt response to exactly what it is you require assistance with.

Blackboard Course Site & Navigation:
For this course I have designed our blackboard course site to include a folder for each week (1 – 14). These folders may be found in our **"Course Documents"** tab. In each folder you will find a TO DO LIST for the week as well as extra materials. Please, take the time during the first week to familiarize yourself with our course site.


TO DO LIST: Week [redacted]

Enabled: Statistics Tracking

Topic		Points
Review Instructor Videos		N/A
Review Instructor PowerPoints		N/A
Download Source Code Examples		N/A
Read Chapter[redacted]		N/A
Discussion Board[redacted]		25
Assignment[redacted]		100
Review / Practice / Complete Recursion Activity Sheet		N/A
Possible Weekly Points		Total: 125

Listed Topics:
1. Unified Modeling Language (UML)
2. Object-Oriented design review
3. Inheritance, polymorphism, class hierarchies and abstract classes
4. Advanced Java GUI applications (Using Java FX)
5. Software engineering principles/error and exception handling
6. Algorithm efficiencies
7. Recursion techniques
8. Multi-dimensional arrays using primitives and objects
9. **Data structures**
10. Java API and collection classes (+External APIs)
11. File input and output streams
12. Multi-Threading
13. Standards of professional behavior and ethics
14. Lists
15. Problem solving

Learning Outcomes:
1. Unified Modeling Language (UML)
2. OOP design review
3. Inheritance, polymorphism, class hierarchies and abstract classes
4. Advanced Java GUI applications
5. Software engineering principles/error and exception handling
6. Algorithm efficiencies
7. Recursion techniques
8. Multi-dimensional arrays using primitives and objects
9. Data structures
10. Java API and collection classes
11. File input and output streams
12. Multi-threading
13. Standards of professional behavior and ethics

Teaching Methods:

This is a 14-week course which will be split into 14 modules to be completed each week.
Please make sure to familiarize yourself with our blackboard course structure the first week of class

Each module's deadline is 11:59pm on the day listed on Blackboard
A new module will open each Saturday and close each Monday (9-Day periods).

This is an online Internet course. Students will meet the course objectives through written lectures, instructor provided videos, exercises, textbook and online readings, assignments, case studies, and use of the Blackboard discussion board. Students can expect to spend at time reading and performing the exercises and assignments of the course.

- Class sessions will consist of:
  - Instructor provided videos
  - Instructor source code Examples (Similar to lab assignments/exams)
  - PowerPoint slide
  - Handout examples
  - Pseudocode, use-case & UML diagram exercises
  - Correspondence/assistance via email and discussion board posts
  - Programming exercises (Hands-on Exercises & Challenges)
- Organization is key to being successful in this course. Organize files in folders. By employing this strategy, referencing examples to problem solve is made much easier. I cannot emphasize the need to organize.

Evaluation Plan:

The course consists of a combination of **5** Lab Assignments, **7** Discussion Board Posts, 1 Mini-Lab Assignment, **1** Final Project.

Assessment	Points (Per)	Total Points
Lab Assignments (5)	120 Points	600
Discussion Board (7)	20 Points	140
Walkthrough Lab Assignment (1)	60 Points	60
Final Project	200 Points	200
		1000

Assignments will require additional time outside of class. Expect to spend time each week reading resources and writing programs

Lab Assignments: Your **5** Lab assignments account for a total of **60%** towards your final grade. These lab assignments will always be based on topics we have already discussed in class. Not completing these assignments is the quickest way to fail this course. Each lab assignment will have a due date. Lab assignments need to be submitted to Blackboard by the listed due date and time. Assignments will not be accepted past the deadline and a zero is earned for that assignment. All lab assignments will be graded after the due date.

Discussion Board Posts: Although we are online course, I would like to create as much of a classroom atmosphere as possible. It is my belief that communication amongst classmates, even online can only benefit our learning. Each post will be relevant to the topic that week or current assignment and in turn generate a conversation which will benefit everyone involved. For this reason, I have created a total of **7** (Usually between assignment weeks) which count towards **14%** towards your overall grade. Discussion board posts throughout the course of the semester in which you will communicate with classmates about listed topics.

Walkthrough Lab Assignment: There will be 1 "Mini" Walkthrough Lab Assignment which will be worth half the points of a regular assignment, **6%** towards you overall grade. This assignment will be heavily guided by instructor for student to follow along and complete. This was designed to counter overlapping of assignments during later weeks of the class.

Final Project: In place of a final exam, I have decided to go with a final project this semester. The details of your final project will be discussed nearer the time, but it will be an application that ties into to a lab assignment with the creation of appropriate documentation. This project will be worth **20%** of your overall grade.

Your final grade will be based on the follow scale:

Grade	Percentage of Points Earned
A	90% or Higher
B	80% to 89%
C	70% to 79%
D	60% to 69%
F	Less than 60%

Other Policies and Procedures:

- This course is a hands-on advanced level computer programming course which is fast paced. You will find this course to be analytical. It will require algorithmic, logical thinking as well as a base knowledge of OOP concepts. **Questions pertaining to current course material are encouraged.**

- As this is an online course, there is no class session to attend at a certain date/time. You are welcome to work on the coursework at a day & time that best fits your schedule**, but you must pay attention to due dates and submit work on time**. Please start your work as early as possible to maximize the time not only to complete assignments, but to ask for help! Blackboard activity will be tracked.

- Emails & Response time: I am more than happy to answer questions via email, the earlier you ask the more assistance I can provide. Though in most cases I will respond much quicker, please allow up to 24 hours for a response depending on when the email is sent. I.e., a message sent at 2AM will not receive an instant response. For this reason, emails sent regarding assignments 24-hours or less before the due date do not guarantee responses before the deadline. Please ask your questions early to maximize help and be as detailed as possible in your question, referencing line numbers and specifics when possible.
  Refer to page 2 *Email etiquette for more information.*

- Please do your own lab assignments. If you find working with another classmate helpful, then change the names of the identifiers used in your program assignments from your teammate's code. A zero is earned for the lab assignment for not following these directions for all team members.

- Punctuality, it is important that you meet deadlines in this course. Many of the assignments are spread out over the course of the semester to allow you time to not only complete each assignment, but also ask questions! This also applies to lectures, please arrive at our scheduled time if you are listed for an in-person section of the course.

- It is important to read the textbook chapters before the material is covered in class. Students find they may need to read the material several times. Time outside of class is needed to write programs in order to gain a solid understanding of the material covered in this course. In class is not the time to listen to music, text, or complete lab assignments while new material is being discussed.

- **Plagiarism and Cheating:** All assignments, projects, test and papers are expected to be done independently and should be the original work of the student. When another source is used, it must be cited as a reference or resource. Cheating and academic dishonesty are very serious offenses. If you have questions about what does or does not constitute plagiarism (knowingly representing the words, ideas, or data of another – whether published or not – as one's own in any academic exercise), please contact me.

  Anyone cheating on an exam or copying work from another source without proper acknowledgement will receive a failing grade in this class. (Refer to Pages 20 – 21 of the CCAC Student Handbook, portions of which have been appended to this course outline.)

- All students are expected to read and comply with the policies and regulations set forth in the CCAC Student Handbook, including without limitation the College's policies regarding academic and behavioral conduct, the procedures for requesting an accommodation based upon a disability, pregnancy or pregnancy related condition, or a religious observance, and for reporting unlawful discrimination and harassment.

- The Student Handbook is available to view and download from the College's website at the following URL: https://www.ccac.edu/policies

- The full text of the College's *Policy Manual, Administrative Regulations Manual*, and the Civil Rights Complaint Procedure can also be viewed and downloaded at: https://www.ccac.edu/policies

- Information concerning the process and documentation required to request a disability-related accommodation can be obtained by contacting the campus' Office of Supportive Services for Students with Disabilities (OSSSD) or by visiting the OSSSD information page at https://www.ccac.edu/policies

- Students are reminded that they can access their course information and CCAC email account, the CCAC Academic Calendar (including add/drop/withdrawal deadlines), the Student Handbook, the College's Incident Report form, and many other College services through the MyCCAC portal at: https://my.ccac.edu

- **Please feel free to talk to me about course issues that concern you.** The policies and regulations outlined in the Student Handbook will be followed. Please refer to the section, Student Code of Conduct, in your Student Handbook

Important Dates:

Jan 16	Martin Luther King Day Observance – No Class
Jan 17	16 Week Classes Begin
Jan 30 (Monday)	**14 Week Classes Begin (Our Class Starts)**
Apr 3 – April 9	Spring Break
Apr 18	Final Day to Withdraw With "W" Grade (For 14-week classes)
May 7	**Classes End**
May 17	Final Grades Posted

Students with Disabilities:
The Community College of Allegheny County makes every effort to provide reasonable accommodations for students with disabilities. Questions about services and procedures for students with disabilities should be directed to the Office of Supportive Services at your campus.

Course Outline Corrections:
During the semester/session, reasonable changes to the course outline may be academically appropriate. Students will be notified of these adjustments by the instructor in a timely manner.

During the semester/session, reasonable changes to the course outline may be academically appropriate. Students will be notified of these adjustments by the instructor in a timely manner

Course Plan: CIT-244 OL **This course schedule is subject to change. Students will be notified of any changes**

Class Week/Date	Lesson or Topic	Learning Activities	Assignments
(1) 01 / 30	OOP Review (Basics)	CIT-130 Review / Introduction to OOP (Object oriented programming) concepts.	-Discussion Board 1 -Assignment 1: Questionnaire
(2) 02 / 04	OOP Review (Basics & UML) Continued	CIT-130 OOP Review cont. Learn how to read & create UML diagrams, the object type, and more.	-Assignment 2 (OOP)
(3) 02 / 11	Inheritance, Abstract Classes & Interfaces	CIT-130 Review Part 2 learn & develop programs dealing with Inheritance, Abstraction & Interfaces	-Discussion Board 2
(4) 02 / 18	Arrays (1D & 2D), Algorithms, Solving Problems with Arrays + (Brief refresher on File I/O)	-Solving problems with arrays: Reverse Array & More -2D Tic tac toe game! -Reading & Writing data	-Discussion Board 3
(5) 02 / 25	Solving Problems Cont.	Start to really think about problem solving. How to approach and use algorithms to solve problems (Encryption, Decryption & More)	-Assignment 3 (Algorithms)
(6) 03 / 04	Recursion	-Learn the basics of recursion through several in class examples. This will set the foundation for when we learn various sorting algorithms. -Indirect VS. Direct recursion	-Discussion Board 4
(7) 03 / 11	Sorting Algorithms	Discuss, write, test, and implement several different **sorting** algorithms. Compare complexities and test on different data sets.	-Discussion Board 5
(8) 03 / 18	Searching Algorithms	Discuss, write, test, and implement several different **searching** algorithms. Compare complexities and test on different data sets.	-Assignment 4 (Searching & Sorting)
(9) 03 / 25	Lists(1) (Wrapper Classes Review) - Array Lists & Generics *(Data Structures Part 1)*	-Data Structures - Learn what a list is (Dynamic Arrays) -How to use Javas API to work with Lists -Generics (What & Why)	
	Lists(2) Lists cont. Linked Lists	-Implementing the Linked List data structure from scratch	
04/03 - /04/09	Spring Break		
(10) 04 / 08	Stacks & Queues *(Data Structures Part 2)*	-An overview of the Stack & Queue data structures through examples -Implement our own Stack & Queues from scratch	-Walkthrough Lab Assignment (Stacks, Generics & Algorithms)
(11) 04 / 15	APIs	-Take a look at some external APIs, learn how to: *Send text messages, Scape website data & more using Java*	-Discussion Board 6 -Assignment 5 (API)
(12) 04 / 22	Graphical User Interfaces FX (Review & Basics)	-Review of 130 GUI concepts -Continue with more GUI concepts	-Final Project Assigned (Java FX GUI)
(13) 04 / 29	Graphical User Interfaces (Advanced Features & Ideas) & Designing an API	-Advanced GUI Concepts -Thinking about writing programs other people / programmers may use	
(14) 05 / 06	Wrapping Up / Finals Week *(Shorter week)*		-Discussion Board 7 -Final Project Due