CS 213: Object Oriented Programming
Fall (Sept-Dec) 2025
Syllabus

Instructor (Cohort A): *Dr. Daniel Addo* *Office*: Room 205 (Engineering Building) *Email*: danieladdo@ashesi.edu.gh *Office hours:* TBD	**Faculty Intern (Cohorts A, B & C)** Mr. John Adenu Mensah Office: Room 107 (Engineering Building) Email: *Office hours:* TBD
Instructor (Cohort B): Dr. David Ebo Adjepon-Yamoah *Office*: Room 205-D (Engineering Building) *Email*: dadjepon@ashesi.edu.gh *Office hours: Tuesdays 12:30 – 14:00*	**Faculty Intern (Cohorts A, B & C)** Mr. Hutton Addy Office: Room 107 (Engineering Building) Email: hamison-addy@ashesi.edu.gh *Office hours:* TBD
Instructor (Cohort C): *Dr. Daniel Addo* *Office*: Room 205 (Engineering Building) *Email*: danieladdo@ashesi.edu.gh *Office hours:* TBD	

Meeting Times:

Lectures (Location: Nana Araba Apt 217):

Cohort A: Tuesdays & Thursdays 08:00 – 09:30
Cohort B: Tuesdays & Thursdays 09:45 – 11:15
Cohort C: Tuesdays & Thursdays 11:30 – 13:00

Labs (Location: Nana Araba Apt 217):

Cohort A: Fridays 12:15 - 13:45
Cohort B: Fridays 13:55 - 15:25
Cohort C: Fridays 10:20 - 11:50

Course Overview:

This course builds upon the programming concepts from the Computer Programming for CS course and will develop students' ability to programme using the object-oriented paradigm and the Java language. It will give students an appreciation of the advantages of object-oriented programming, help them define and construct objects; and leverage abstraction, inheritance, polymorphism, and encapsulation to develop robust and maintainable applications. It will also introduce students to event-driven programming and graphical user interfaces, as well as the use of standard Java packages and the Java collections API. Emphasis will be placed on good coding practices including naming conventions, documentation and testing. The course also introduces the concept of Unified Modeling Language (UML) for object-oriented design. Through one or more team projects, students will gain experience in designing and implementing larger applications than in their previous programming classes.

Course Learning Outcomes:

At the end of the course, students will be able to:

1. Demonstrate the ability to write syntactically correct Java code that solves given problems.
2. Apply the principles of object-oriented programming in building Java applications.
3. Understand and make use of the UML notation for object-oriented analysis and design.
4. Create basic graphical user interfaces using JavaFX.
5. Use standard Java libraries and APIs for various tasks such as file-handling, networking and collections.
6. Able to design, develop and document simple Java-based applications for real-world solutions in a variety of domains such as business, science and society.

Ashesi Learning Goals Addressed in this Course:

1. **Ethics and Civic Engagement:** *An Ashesi student is an ethical, responsible and engaged member of his/her community*. Students are expected to maintain a very high academic and ethical standard, as described in the "Expectations" section below.
2. **Critical Thinking and Quantitative Reasoning**: *An Ashesi student is able to apply critical thinking and quantitative reasoning to approach complex problems.* This course is all about problem-solving with computers. Students will develop the ability to analyze relevant problems, design algorithms to solve them, and implement these algorithms in the Java programming language.
3. **Communication**: *An Ashesi student is an excellent communicator in a variety of forms*. This course requires students to write short 1-paragraph reflections of assignments and projects.
4. **Curious and Skilled**: *An Ashesi student is inquisitive and confident, has breadth of knowledge, and has attained a high level of mastery in their chosen field.* This course aims to develop skilled problem-solvers and programmers who have a widening understanding of the role of computer science and information systems in our complex world.
5. **Technology Competence**: *An Ashesi student is an effective and flexible user of technology*. This course focuses on developing a particular aspect of technology competence, namely programming in Java.

Textbooks & Resources:

Primary Textbook

Liang, Y. D. (2024). *Introduction to Java programming and data structures* (13th ed.). Pearson. ISBN: 9780134670942 (Introduction to Java Programming and Data Structures, Global Edition - BUKU)

Supplementary Texts

Savitch, W. (2011). *Java: An introduction to problem solving and programming* (6th ed.). Pearson. ISBN: 9780132162708

Savitch, W., & Mock, K. (2015). *Absolute Java* (6th ed.). Pearson. ISBN: 9780134041674

Oracle. (n.d.). *Java platform, standard edition API specification*. Retrieved from Java JDK 24 (Overview (Java SE 24 & JDK 24)

JavaFX Resources

Lowe, D. (2014). *JavaFX for dummies*. Wiley. ISBN: 9781118385340

Dea, C., Heckler, M., Grunwald, G., Pereda, J., & Phillips, S. (2014). *JavaFX 8: Introduction by example* (2nd ed.). Apress. ISBN: 9781430264606

Coursera. (n.d.). *Starting GUI programming with JavaFX*. Retrieved from https://www.coursera.org/projects/gui-programming-javafx

Learning Activities:

In this course, learning takes place through a variety of activities, including lectures, reading, labs and assignments. Each of these is carefully designed to help you develop expertise in problem-solving and programming, and none should be neglected. Historically, students who have paid careful attention to each of these aspects have been very successful and students who have neglected one or more of them have struggled.

- *Lectures*: The lectures will present essential concepts and include activities to help you grasp the essential principles and approaches that will make you a good problem-solver and programmer. The best way to get the most of the lectures is to be an active participant: coming to class well prepared, asking and answering questions in class, and participating in any discussions.
- *Reading*: The assigned reading introduces new concepts and clarifies and expands upon the material discussed in class. It includes additional details that we will not have time to dwell on in class, so you must do the reading in order not to miss this information.
- *Labs and Assignments*: Labs and assignments are primarily designed to give you practice and help you learn concepts that you cannot learn just by listening to a lecture. It is an opportunity to apply concepts in a context that allows you to seek

clarification and guidance from the instructor and teaching assistants. Students who ensure that they complete and understand all the assignments typically go on to do well in the tests and exams, whereas students who fail to complete assignments typically go on to do very poorly in the exams.

Expectations:

The instructor and teaching assistant are committed to helping you to be successful in this course. In return, there are some fundamental expectations of you.

Participation
Your active participation enriches the course experience for everyone. Do not be afraid to ask questions! Your questions will probably help others in the class as well.

Professionalism
You are expected to interact with your course colleagues, as well as the instructor and teaching assistant in a professional and polite manner always.

Academic honesty
You are expected to always keep in mind that *"An Ashesi student is an ethical, responsible and engaged member of his/her community"*. The work in this course is designed to help you develop skills essential to your future career success. You can only develop these skills if you do the work yourself. All the work that you turn in *must* be your own. For programming assignments, you are allowed and encouraged to brainstorm about the problems with your peers. You can talk in English (not Java!) about algorithms and approaches. However, unless you are explicitly asked to work in a team, you must sit down behind the computer and write the program you turn in yourself. If you copy a section of code from any source (e.g., part of an example from class, or from the course textbook) you must include an appropriate citation in a comment above the code segment that you copied. You may not copy code from the internet or from your peers – this is plagiarism, and furthermore, will not help you learn.

Evaluation Criteria
Your grade in the course will be determined according to the following criteria:

Class Preparation Assignments & Quizzes:	10%
Programming Assignments:	25%
Mid-Semester Exam:	20%
Final Project:	10%
Final Exam:	30%
Attendance & Participation:	5%

Some assignments may have some "extra credit" options for those who would like an additional challenge. The most important outcome of extra credit work is extra learning. Furthermore, doing extra credit work *may* influence your final grade if you are borderline (within 1% of a grade boundary) at the end of the semester. However, not doing extra credit work cannot harm you. Note that you should not do extra credit work at the expense of the main assignment, because it is the grade on the main assignment that counts towards your base grade for the course.

Late Policy

Class preparation assignments and quizzes are due as stated. No late class preparation assignments or make-up quizzes will be accepted for any reason, but your lowest grade will be dropped. All programming assignments are due when stated. You will lose 10% of your grade for each **day** your assignment is handed in late.

Schedule

Below is a tentative schedule. The schedule is subject to adjustment at the instructors' discretion. All updates will be posted on CANVAS.

Week	**Topics**	**Reading**	**Assignments & Assessments**
Week 1 Sept 8 – 12	Introduction to Object-Oriented Programming Introduction to Java: • Basic syntax & conventions • Selection / branching	Liang Ch 1, 2 & 3	Assignment 1 released
Week 2 Sept 15 – 19	Mathematical functions, characters, strings, and loops	Liang Ch 4 & 5	**Assignment 1 due** (Thur, Sept 18)
Week 3 Sept 22 – 26	Methods and arrays	Liang Ch 6 & 7	Assignment 2 released (Tue)
Week 4 Sept 29 – Oct 3	Object-oriented analysis and design	Liang Ch 9	**Assignment 2 due** (Thur, Oct 2)
Week 5 Oct 6 – 10	More on objects and classes	Liang Ch 10	Assignment 3 released (Tue)
Week 6 Oct 13 – 17	Polymorphism & inheritance	Liang Ch 11	**Assignment 3 due** (Thur, Oct 16)
Week 7 Oct 20 – 24	More on polymorphism & inheritance	Liang Ch 13	Assignment 4 released (Tue)
Week 8 Oct 27 – 31	Mid-semester review and exam		**Mid-Sem Exam** (Fri, Oct 31)
Week 9 Nov 3 – 7	MID-SEMESTER BREAK		
Week 10 Nov 10 – 14	Exceptions and exception handling Introduction to Java FX & GUIs	Liang Ch 12 & 14	**Assignment 4 due** (Thur, Nov 13) **Project Initiated**
Week 11 Nov 17 – 21	File I/O Optional: Network I/O	Liang Ch 12 & 15	Project Milestone 1 (Thu, Nov 20)

	Event-driven programming		
Week 12 Nov 24 – 28	Java Collections Framework	Liang Ch 19 & 20	Project Milestone 2 (Thu, Nov 27)
Week 13 Dec 1 – 5	Java Collections Framework Class-wrap	Liang Ch 21	Project Submission (Fri, Dec 5)
Weeks 14 Dec 8 – 12	Project Presentations		Tue, Thur & Fri
Weeks 15 Dec 8 – 12	EXAMS		

Artificial Intelligence (AI) Policy

The use of AI can add valuable experiences in real life. We can benefit from a wide range of sample data and problem-solving. We can take advantage of these models to improve our learning skills and knowledge. AI tools can help us to personalize our learning experiences by adapting to the individual's needs. For example, ChatGPT opened a new way to find answers to our questions. It also can assist us in programming. It can be one of the tools that can empower you to write larger and more powerful software. But it also raises concerns about integrity, privacy, biases, and much more. The potential drawbacks are that it can hinder the ability to develop foundational programming skills, deter the ability to develop problem-solving skills, limit creativity, reduce learning engagement, and lead to legal and ethical issues. Therefore, you may use AI models (such as ChatGPT) to help study and understand class materials, but you are not allowed to use them for your assignments, projects, and quizzes. You may not put assignment prompts into generative AI tools, and you may not submit any AI-generated work for grading. Violations of these guidelines will be considered as academic integrity violation and plagiarism.