

CS120 Intro to OOP, Section C, Spring 2020

Term/Year: Spring 2020
Subject Code and Course Number: CS120, Section C
Course Title: Introduction to Object-Oriented Programming
Number of Credits: 3
Instructor Name: Varduhi Yeghiazaryan
Email Address: vyeghiazaryan@uaa.am
Telephone Number: 060 612633
Office Location: 331W
Office Hours: Tuesday and Friday 15.30–16.30,
or by appointment

Teaching Associate: Yeva Gabrielyan
TA Problem Solving Session: Lab 003, Wednesday 17.30–19.30
TA Email Address: yeva_gabrielyan19@alumni.uaa.am
TA Office: 310W
TA Office Hours: Monday 17.30–19.00

Class Schedule: 313W, Tuesday and Thursday 12.00–13.15

Course Description: The course will survey the following topics: control structures, functions, arrays, strings, introduction to UML, classes and data abstraction, inheritance, introduction to polymorphism, abstract classes and interfaces. Students are required to develop basic proficiency in utilizing and testing software systems related to computational sciences and in applying at least one programming language to software development. Three hours of instructor-led class time per week including discussions and problem sets.

Subject to Change to accommodate student needs.

Prerequisites: CS110 Introduction to Computer Science
Co-Requisites: None

Required Materials

Main textbooks:

Walter Savitch. Absolute Java, 6th edition, Pearson, 2016.

Walter Savitch. Absolute C++, 6th edition, Pearson, 2016.

Additional books:

Paul Deitel and Harvey Deitel. Java How to Program, 11th edition, Pearson, 2018.

Herbert Schildt. Java: A Beginner's Guide, 8th edition, McGraw-Hill Education, 2018.

Paul Deitel and Harvey Deitel. C++ How to Program, 10th edition, Pearson, 2017.

Online Resources:

The Java™ Tutorials <https://docs.oracle.com/javase/tutorial/>

Java™ Platform, Standard Edition & Java Development Kit Version 13 API Specification
<https://docs.oracle.com/en/java/javase/13/docs/api/>

C++ Language Tutorials <http://www.cplusplus.com/doc/tutorial/>

Standard C++ Library Reference <http://www.cplusplus.com/reference/>

C++ Reference <https://en.cppreference.com/w/>

Technology:

Computer with JDK8 or later version, IntelliJ IDEA or Eclipse IDE, Visual Studio

CS120 Intro to OOP, Section C, Spring 2020

Schedule & Topics:

Course Syllabus is subject to change to address student needs.

Week	Class	Topic	Reading	Non-Reading Home Tasks
1	1	Introduction to Java, variables	AJ 1.1, 1.2	HW1 release
1	2	Data types, operators, program style	AJ 1.2, 3.2, 1.4	
2	3	Strings, input and output	AJ 1.3, ch2	
3	4	Branching mechanisms, loops	AJ 3.1, 3.3	HW2 release
3	5	Arrays, MD arrays	AJ 6.1, 6.4	
4	6	Methods, parameters	AJ 4.1, 4.3, 6.2	
4	7	Recursion, debugging, random numbers	AJ ch11, 3.4, 3.5	
5	8	Objects and classes, information hiding and encapsulation	AJ 4.1, 4.2	HW3 release
5	9	Constructors, statics, references	AJ 4.4, 5.1, 5.2	
6	10	References, packages, documentation	AJ 5.2, 5.3, 5.4	
6	11	Summary, Q&A		
Mid-Term Exam 1				
7	12	Review of Mid-Term Exam 1		
7	13	Inheritance	AJ 7.1, 7.2	HW4 release
8	14	Object, for-each, enum	AJ 7.3, 6.3	
8	15	Polymorphism, abstract classes	AJ ch8	
9	16	Exception handling	AJ 9.1, 9.2	HW5 release
9	17	Exception handling, UML	AJ 9.3, 12.1	
10	18	File I/O	AJ 10.1, 10.2, 10.3	
10	19	Interfaces, inner classes	AJ ch13	
11	20	ArrayList, generics	AJ ch14	HW6 release
11	21	GUI	AJ ch17	
12	22	GUI	AJ ch18	
12	23	Summary, Q&A		
Mid-Term Exam 2				
13	24	Review of Mid-Term Exam 2		
13	25	C++		HW7 release
14	26	C++		
14	27	C++		
15	28	C++		
15	29	Summary, Q&A		

Topic-specific reading will be assigned at the end of each class.

CS120 Intro to OOP, Section C, Spring 2020

Student Learning Outcomes:

The following chart shows alignment between course-specific and program student learning outcomes and program goals.

Program Goal	Program Student Learning Outcomes <i>Students will be able to:</i>	Course-based Student Learning Outcomes <i>In this course, students will be able to:</i>
1. Equip students with knowledge and advanced skills in mathematical reasoning, problem solving, modeling and scientific computation	1.3 Utilize and adapt software and select and use hardware systems related to computer science (Beginner Level)	<i>Solve modestly sized problems applying the principles of object-oriented programming.</i>
3. Prepare students for development of scientific, engineering and industrial software applications	3.2 Develop and implement software applications in one or more programming languages (Beginner Level) 3.3 Develop and test software tools and methods (Beginner Level)	<i>Design and implement well-structured console and GUI applications in Java using object-oriented design.</i>
5. Provide students with a broad foundation of knowledge and skills and cultivate a commitment to life-long learning	5.1 Use common software and information technology to pursue inquiry relevant to their academic and professional fields, and personal interests (Advanced Level)	<i>Use integrated development environments for software development.</i>

Course Structure: Instructor-led class will meet twice per week. Students are strongly encouraged to attend these classes, as well as weekly problem solving sessions. Home tasks include reading and problem sets with programming assignments.

Method of Evaluation: Student progress will be assessed by homework assignments, participation, mid-term exams and a final exam.

Homework: Homework assignments will be given on a weekly or bi-weekly basis except for the last and exam weeks. You are encouraged to study and discuss problems with fellow students, but you must write and code your own homework by yourself, and make sure you understand how to obtain the solutions to the problems. Any collaboration will always be graded zero. Homework problems will be assigned by the instructor throughout the semester and will be posted on Moodle. Students are to submit their work electronically before the deadline. Late homework will not be graded. The format of submitting homework assignments will be separately posted on Moodle.

Class attendance and participation: Students are expected to attend class and demonstrate their understanding of topics by participating in class discussions. Participation assessment will be mainly based on 4–6 quizzes, attendance and participation in classroom discussions. The quizzes will be closed-book and closed-device: use of calculators, computers, tables, phones are prohibited; books and notes may not be consulted. Collaboration during quizzes is strictly forbidden.

CS120 Intro to OOP, Section C, Spring 2020

Mid-Term Exams: The course will include two mid-term exams. The midterms are closed-book and closed-device: use of calculators, computers, tablets, phones is prohibited; books and notes may not be consulted. Collaboration during exams is strictly forbidden.

Final Exam: The final exam will cover all topics covered throughout the course. The purpose of the final exam is to assess students' mastery of concepts and terminology as well as their abilities to select appropriate methods and apply this knowledge to solve complex practical problems.

Moodle: You should enroll into the course on Moodle. Homework assignments, learning materials and students' grades will be posted regularly on Moodle. Thus, please check the course's Moodle page regularly.

Assessment: Student learning will be evaluated on the basis of the following weighted components:

- Homework (21%)
- Participation (10%)
- Mid-Term Exams (34%)
- Final Exam (35%)

The final grades will be defined according to the following ranges given below.

Grade	Grade Point	Percentile Range
A+	4	95 – ≤100
A	4	90 – < 95
A-	3.7	85 – < 90
B+	3.3	80 – < 85
B	3	75 – < 80
B-	2.7	70 – < 75
C+	2.3	66 – < 70
C	2	62 – < 66
C-	1.7	58 – < 62
D+	1.3	55 – < 58
D	1	53 – < 55
D-	0.7	50 – < 53
F	0	0 – < 50

Library and Media/Technology Use

Students are encouraged to use supplemental online and reference materials available at the library to enhance their overall learning in the course. If students have any questions or need additional support in using library resources or technology, they should confer with library staff, ICT, or the instructor.

Late Policy

Students are required to submit homework assignments by the deadline scheduled by the instructor. Homework submitted late will not qualify for grading.

Make-up Procedures

Make-up assessment for missed exams can be organized only in exceptional cases at the instructor's discretion. The student must submit convincing evidence of a medical or other emergency that makes taking the exam at the scheduled time impossible. It is also important to notify the instructor in advance in cases when taking exam on time is impossible for whatever reasons.

Policy on Grade Appeal

Students are entitled to appeal grades in line with the university's *Grade Policies* policy which is available online at <http://policies.aua.am/policy/11>

Standards for Academic Integrity

Students are required to conduct themselves in an academically responsible and ethical manner in line with AUA's *Student Code of Ethics*. Acts of academic dishonesty impair the academic integrity of AUA and create an unfair academic advantage for the student involved and other member(s) of the academic community. These acts are subject to disciplinary measures as prescribed in the AUA Student Code of Ethics, <http://policies.aua.am/policy/10>

The Student Code of Conduct can be found at <http://policies.aua.am/policy/101>

Special Needs:

Students requiring special accommodations for learning should contact the Center for Student Success by the end of the Add/Drop period with such requests. studentsuccess@aua.am, <https://studentsuccess.aua.am/disability-support-services/>