





Computer Science (COMP) 268

Introduction to Computer Programming (Java) (Revision 12)

Status:

Replaced with new revision, see the [course listing](#)  for the current revision 

Delivery mode:

Individualized study online 

Credits:

3

Area of study:

Science

Prerequisites:

COMP 200 or Course Coordinator approval.

Precluded:

None

Challenge:

COMP 268 has a challenge for credit option.

Faculty:

Faculty of Science and Technology 

Notes:

Students who are concerned about not meeting the prerequisites for this course are encouraged to contact the [course coordinator](#) before registering.

Overview

COMP 268 is designed to introduce you to programming in the Java computer language. The course progresses from first principles to advanced topics in object-oriented programming using Java.

The course builds on the basic elements of Java programming and culminates in a final project (Assignment 4) that incorporates the skills acquired in the course: a text-based adventure game.

Students are also required to reflect on their progress and learning throughout the course, and to engage in online discussions with their peers, in an effort to increase problem-solving and critical analysis skills.

Outline

The course consists of the following Study Guide units, which are based on their corresponding textbook chapters. Units denoted with an asterisk (*) are supplementary reading and are considered outside the scope of the course:



- Unit 1: Overview: The Mental Landscape
- Unit 2: Programming in the Small I: Names and Things
- Unit 3: Programming in the Small II: Control
- Unit 4: Programming in the Large I: Subroutines
- Unit 5: Programming in the Large II: Objects and Classes
- * Unit 6: Introduction to GUI Programming
- Unit 7: Arrays and ArrayLists
- Unit 8: Correctness, Robustness, Efficiency
- * Unit 9: Linked Data Structures and Recursion
- * Unit 10: Generic Programming and Collection Classes
- Unit 11: Input/Output Streams, Files, and Networking
- * Unit 12: Threads and Multiprocessing
- * Unit 13: GUI Programming Continued

Learning outcomes

Upon successful completion of this course, you will be able to

- articulate the principles of object-oriented problem solving and programming.
- outline the essential features and elements of the Java programming language.
- explain programming fundamentals, including statement and control flow and recursion.
- apply the concepts of class, method, constructor, instance, data abstraction, function abstraction, inheritance, overriding, overloading, and polymorphism.
- program with basic data structures using array, list, and linked structures.
- explain the object-oriented design process and the concept of software engineering.
- program using objects and data abstraction, class, and methods in function abstraction.
- analyze, write, debug, and test basic Java codes using the approaches introduced in the course.
- analyze problems and implement simple Java applications using an object-oriented software engineering approach.

Evaluation

To **receive credit**  for COMP 268, students must score at least **D (50 percent)**  in each of the following assessment activities. Assignments 1 through 4 require the student to write programs in Java and document their reflections on their work. A participation mark is awarded based on student involvement in online discussions.

Activity	Weight
Assignment 1	15%

Activity	Weight
Assignment 2	15%
Assignment 3	15%
Assignment 4	25%
Participation	15%
Final Exam	15%
Total	100%

The **final examination** for this course must be requested in advance and written under the supervision of an AU-approved exam invigilator. Invigilators include either ProctorU or an approved in-person invigilation centre that can accommodate online exams. Students are responsible for payment of any invigilation fees. Information on exam request deadlines, invigilators, and other exam-related questions, can be found at the [Exams and grades](#) section of the Calendar.

To learn more about assignments and examinations, please refer to Athabasca University's [online Calendar](#).

Materials

This course either does not have a course package or the textbooks are open-source material and available to students at no cost. This course has a **Course Administration and Technology Fee**, but students are not charged the Course Materials Fee.

Digital course materials

Links to the following course materials will be made available in the course:

Eck, D. J. (2019, July). [Introduction to Programming Using Java](#) (Version 8.1).

Other Materials

All course learning materials for COMP 268, including Java compilers and web editors, are distributed in electronic format through the course site and other websites.

Special Course Features

COMP 268 is offered through Moodle, a learning management system that is accessed through the myAU portal. COMP 268 can be completed at the student's workplace or home. COMP 268 is an elective in all undergraduate programs offered by the School of Computing and Information Systems.

Challenge for credit

Overview

The challenge for credit process allows you to demonstrate that you have acquired a command of the general subject matter, knowledge, intellectual and/or other skills that would normally be found in a university-level course.

Full information about **challenge for credit** [↗](#) can be found in the Undergraduate Calendar.

Evaluation

To **receive credit** [↗](#) for the COMP 268 challenge registration, you must achieve a grade of at least B (75 percent) on the examination and **B (75 percent)** [📄](#) on EACH assignment.

[📄](#) **Challenge for credit course registration form**

Important links

- › [Academic advising](#) [↗](#)
- › [Program planning](#) [↗](#)
- › [Request assistance](#) [↗](#)
- › [Support services](#) [↗](#)

Athabasca University reserves the right to amend course outlines occasionally and without notice. Courses offered by other delivery methods may vary from their individualized study counterparts.

Opened in Revision 12, March 5, 2020

Updated April 14, 2025

View **previous revision** [↗](#)
