

Course Outline

Course Code, Number, and Title: CPSC 1181: Object-Oriented Computing

Year and semester: Spring 2023

Course Format: Lecture 4.0 h + Seminar 0.0 h + Lab 2.0 h

Credits: 3

Transfer Credit: For information, visit bctransferguide.ca

Course Description:

Object-oriented programming (OOP) is a paradigm to design and develop software based on the concept of objects. Students are introduced to the fundamental concepts of programming from an object-oriented (OO) perspective: abstraction; objects; classes and class hierarchies; methods; encapsulation and information hiding; inheritance; polymorphism. Students learn and practice the application of OO design with modeling tools (e.g., class diagrams), container/collection classes, event-driven programming, exception handling, GUI, multi-threading, and networking. The focus is placed on good software engineering principles using a language that supports the OO paradigm.

Prerequisite(s): A minimum "C" grade in CPSC 1150 or 1155; or permission of department.
Prerequisites are valid for only three years.

Learning Outcomes:

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

- Design and implement applications using the Object-Oriented Programming (OOP) methodology
- Describe and employ the concepts of OOP, such as encapsulation, inheritance, polymorphism, abstract classes, interfaces, exceptions and exception handling
- Design, draw, and explain Unified Modelling Language (UML) class diagrams
- Build a program that manipulates data using a collection, such as an ArrayList or vector
- Implement programs using generic programming
- Use a given design pattern to develop and implement an Object-Oriented (OO) solution to a given problem
- Use, design, and write unit tests
- Apply the example of object-oriented design to build programs containing a selection of event driven programming, GUI, graphics, multi-threading/race condition/synchronization, network programming, lambdas and streams
- Write internal and external documentation that conforms to language conventions and good programming practice

Instructor(s): Mingwu Chen

Office: B019k

Phone: 604 323 5819

Email: mwchen@langara.ca

Website & Schedules: <http://mylinux.langara.ca/~mwchen>

Office Hours:

TH 12:30-13:20

W 11:30-12:20

Textbook and Course Materials:

1. Big Java: Early Objects, 7th Edition, by Cay S. Horstmann, Wiley

Note: This course may use an electronic (online) instructional resource that is located outside of Canada for mandatory graded class work. You may be required to enter personal information, such as your name and email address, to log in to this resource. This means that your personal information could be stored on servers located outside of Canada and may be accessed by U.S. authorities, subject to federal laws. Where possible, you may log in with an email pseudonym as long as you provide the pseudonym to me so I can identify you when reviewing your class work.

Assessments and Weighting:

Labs	30%
Midterm 1	20%
Midterm 2	20%
Final Exam	30%
Participation	-3% - +3%

Grading:

Letter Grade	Overall Average
A+	≥ 90
A	≥ 85 and < 90
A-	≥ 80 and < 85
B+	≥ 76 and < 80
B	≥ 72 and < 76
B-	≥ 68 and < 72
C+	≥ 64 and < 68
C	≥ 60 and < 64
C-	≥ 55 and < 60
D	≥ 50 and < 55
F	< 50

Detailed Course Schedule:

Week	Topics	Labs	References
Week 1 (Jan 4 - 7)	Introduction to the course	Lab1: Java Essentials	
Week 2 (Jan 8 - 14)	Review on Java essentials	Lab2: JUnit Test	
Week 3 (Jan 15 - 21)	Objects and classes	Lab3: Objects and Classes	Chapter 3
Week 4 (Jan 22 - 28)	ArrayList UML	Lab4: ArrayList and More Classes	Chapter 7
Week 5 (Jan 29 – Feb 4)	Inheritance	Lab5: Inheritance	Chapter 9
Week 6 (Feb 5 - 11)	Object-oriented design Abstract classes and interfaces	Lab6: Interfaces	Chapters 12 & 10
Week 7 (Feb 12 - 18)	Inner class (MT1)		Chapter 10
Week 8 (Feb 19 - 25)	(Spring Break)		
Week 9 (Feb 26 – Mar 4)	JavaFX Basics	Lab7: JavaFX Basics	
Week 10 (Mar 5 - 11)	JavaFX events and layout	Lab8: Event-driven Programming	
Week 11 (Mar 12 - 18)	JavaFX animation Exception handling	Lab9: Animation and Exception Handling	
Week 12 (Mar 19 - 25)	Multithreading (MT2)	Lab10: Multithreading	Chapter 22
Week 13 (Mar 26 – Apr 1)	Networking	Lab11: Networking	Chapter 23
Week 13 (Apr 2 – 6)	Review		

College Policies:

As a student at Langara, you are responsible for familiarizing yourself and complying with the following policies:

[E1003 - Student Code of Conduct](#)

[F1004 - Code of Academic Conduct](#)

[E2008 - Academic Standing - Academic Probation and Academic Suspension](#)

[E2006 - Appeal of Final Grade](#)

[F1002 - Concerns about Instruction](#)

[E2011 - Withdrawal from Courses](#)

Departmental/Course Policies:

- No late assignments/projects will be accepted. No make-up exams. Marks of missed assignments/exams due to medical/similar reasons will be re-allocated to other assignments/exams.
- You may be given an automatic fail if you miss more than 20% of classes without an acceptable reason.
- In order to get a C or higher grade in a computer science course, a student MUST achieve at least 50% of average in the exam components of the course.