

COSC2767E: Object-Oriented Programming

Instructor: Dr. Haibin Zhu **Office:** A124A **Ext:** 4434 **Email:** haibnz@nipissingu.ca
Web: <http://faculty.nipissingu.ca/haibinz> **Office Hours:** Mon: 11:30am-5:00pm or by appointment

Course Description:

It covers the concepts of object-oriented programming (OOP) languages and systems including an introduction to fundamental abstraction, modularity and encapsulation mechanisms in OOP from a software engineering and representational perspective. Basic OOP concepts covered include: polymorphism and operator overloading; message passing via generic functions; late versus early binding times; and inheritance mechanisms and their relationship to the type systems of programming languages. C++ and Java are the two languages used to convey the fundamental concepts and principles. Other topics include: OOP languages versus imperative programming languages; and contrasting pure OOP environments versus mixed paradigm languages.

Prerequisite: COSC 1557, COSC 1567

Recommended Textbook:

Timothy A. Budd, The Introduction to Object-Oriented Programming (3rd Edition), Addison-Wesley, 2001, ISBN 0201760312
 Avinash C. Kak, Programming with Objects: A comparative presentation of object-oriented programming with C++ and Java, John Wiley & Sons, 2003. ISBN: 0-471-26852-6.

Course Objectives:

- This course will introduce the concepts of Object Oriented Programming (OOP) Languages and Systems. The course will start with an introduction of fundamental abstraction, modularity and encapsulation mechanisms in Object Oriented Programming Languages such as C++ and Java.
- Abstract data types will be examined from both a software engineering and representational perspective. Contrasts with imperative programming languages will be made. Advanced C++ and Java constructs such as abstract base classes, static members and other constructs will be presented. Comparisons with Java between C++ will be made.
- Other concepts to be covered include polymorphism, operator overloading, message passing viz. generic functions, late versus early binding times, as well as inheritance mechanisms and their relationship to programming language type systems.
- The emphasis of this course this semester will be the comparison of central features of several OOP. Special topics will be included as time permits. One such topic will be the discussion of Object libraries including the Java class libraries. Pure OOP environments such as Smalltalk will be contrasted with mixed paradigm languages such as C++ and Java.
- Object Oriented Design techniques as well as OOP idiom will be presented. We'll look at classic OOP patterns and frameworks.

Grading:

- | | | |
|-------------------|-------------|------------|
| • Projects: 30% | • A: 100-80 | • D: 59-50 |
| • Tests: 34% | • B: 79-70 | • F: 49 |
| • Final Exam: 36% | • C: 69-60 | |

COSC2767: Object-Oriented Programming (2019WD) (Lectures in A120 at Mon: 8:30pm-11:20pm)

No.	Date	Lecture	Notes
1	01/07	Lect1	Think with objects
2	01/14	Lect2	Object-Oriented Design, Classes and Methods, Project 1
3	01/21	Lect3	Messages, Instances, and Inheritances
4	01/28	Lect4	Subclasses, subtype and Multiple Inheritance, Test 1
5	02/04	Lect5	Static, Dynamic Behavior and Substitution, Project 1 Due, Project 2
6	02/11	Lect6	Polymorphism and overloading,
	02/18	No class	Study week
7	02/25	Lect7	Overriding, Test 2
8	03/04	Lect8	The Polymorphic Variable and Generics, Project 2 due, Project 3
9	03/11	Lect9	Container classes & STL
10	03/18	Lect10	Frameworks and Object Interconnections
11	03/25	Lect11	Patterns, Class Objects and Distributed Objects, Project 3 Due
12	04/01	Seminar	E-CARGO and Review
	04/??	Final Exam	

Note: Tests are in-class, closed books and close notes. **Everything may be subject to change, please pay attention to the class declarations.**