



**Dalhousie University**  
**Faculty of Computer Science**

---

# CSCI 3132 – Object Orientation and Generic Programming

## Introduction

---

# Course Take Away

---

- What is your motivation for taking this course?
  - What do you want to be your take away from this course?
-

# What this course is about

---

- The course comprises of three overlapping parts:
    1. What is object-orientation and how it helps in designing and implementing complex systems. You will learn how to create object-oriented design from a given set of requirements.
    2. Learn object-oriented programming using C++.
    3. Discovering generic programming and how it helps improve modularity and reusability and reduce the amount of code.
  - Note that the three parts are not mutually exclusive and will overlap considerably throughout the course.
-

# Learning Outcomes

---

On successful completion of this course, you will be able to:

1. Identify key concepts of object-orientation: polymorphism, encapsulation, abstraction and inheritance.
  2. Analyze a set of client requirements to identify objects and their relationships and create a high-level design.
  3. Create detailed design based on object-oriented design principles.
  4. Implement features of object-oriented programming using a high-level language such as C++.
  5. Describe the motivation and paradigms of generic programming.
  6. Build generic classes using templates.
-

# Lecture Plan\*

Week	Dates	Lecture Topics
1	06-May to 12-May	Intro to the course. Expressing client requirements
2	13-May to 19-May	Domain modelling, Use case and Class diagrams
3	20-May to 26-May	Object model. OO Concepts
4	27-May to 02-Jun	C++ programming basics, Data types, Pointers review
5	03-Jun to 09-Jun	Function and Operator overloading
6	10-Jun to 16-Jun	Inheritance and Polymorphism
7	17-Jun to 23-Jun	References, Virtual and Pure Virtual functions
8	24-Jun to 30-Jun	In-class Programming Contest
9	01-Jul to 07-Jul	Generic Programming, Templates
10	08-Jul to 14-Jul	Standard Template Library
11	15-Jul to 21-Jul	Group project presentations
12	22-Jul to 28-Jul	Design patterns and Case study
	31-Jul to 06-Aug	FINAL EXAM PERIOD

\* May be subject to minor changes

# Lab Plan\*

Sr. No.	Lab Date	Lab Topics
	06-May	No Lab
Lab 1	13-May	C++ Programming Practice
	20-May	No Lab - Holiday
Lab 2	27-May	Classes and Objects. Inheritance and Composition. Lab Assignment 1
Lab 3	03-Jun	Pointers.
Lab 4	10-Jun	Function and Operator overloading. Lab Assignment 2
Lab 5	17-Jun	Inheritance, Encapsulation and Polymorphism.
Lab 6	24-Jun	Virtual and pure virtual functions. Abstract classes. Lab Assignment 3
	01-Jul	No Lab - Holiday
Lab 7	08-Jul	Templates, Exception handling. Lab Assignment 4
Lab 8	15-Jul	STL.
Lab 9 <sup>#</sup>	22-Jul	Presentations ( <sup>#</sup> if required)

\* May be subject to minor changes

# Important Dates\*

Important Dates	Action
16-May	Quiz 1
27-May	Lab Assignment 1 available
30-May	Quiz 2
04-Jun	Group Project available
09-Jun	Lab Assignment 1 due
10-Jun	Lab Assignment 2 available
13-Jun	Quiz 3
23-Jun	Lab Assignment 2 due
24-Jun	Lab Assignment 3 available
04-Jul	Quiz 4
07-Jul	Lab Assignment 3 due
08-Jul	Lab Assignment 4 available
14-Jul	Group Project due
21-Jul	Lab Assignment 4 due
25-Jul	Quiz 5
31-Jul to 06-Aug	Final Exam

\* May be subject to minor changes

# Text and References

---

- Reference books
    - Bruce Eckel, “Thinking in C++”, Vol 1 & 2, 2<sup>nd</sup> Edition. These can be downloaded free of cost from <http://www.mindviewinc.com/Books/downloads.html>
    - Booch, Maksimchuk, Engle, Young, Conallen, and Houston, “Object-Oriented Analysis and Design with Applications” - 3<sup>rd</sup> Edition, Addison-Wesley, 2011, ISBN-13: 978-0-201-89551-3
    - Gamma E., Helm R., Johnson R., and Vlissides J., “Design Patterns: Elements of Reusable Object-Oriented Software” Addison-Wesley, 1995, ISBN-13: 978-0-201-63361-0
  - Lecture slides and additional course material will be available on the course website on Brightspace as and when required.
-



# Teaching Assistant

---

- Mr. Reza Rahimi

Email: [Reza.Rahimi@dal.ca](mailto:Reza.Rahimi@dal.ca)

Consultation: TBD

Location: Learning Centre

---

# Evaluation Criteria

---

- Quizzes (20%)
    - Five quizzes, best 4 would be considered, each worth 5%
    - Quizzes have to be taken in class only
  - Lab Assignments (20%)
    - Four lab assignments, each worth 5% will be given during lab sessions.
    - Late assignments or assignments that do not compile according to provided instructions will receive a zero grade.
  - Class project and presentation (30%)
    - Class project will be completed by students in groups.
    - Design and implementation of the project will be presented by each group in class
  - Final Exam (30%)
    - The exam will be scheduled by the university.
    - The exam will cover all material in the course
  - Bonus Marks (up to 5%)
    - In-class competition and on-the-spot Q&A
-

# Course Communications

---

- Course announcements will be posted on the course web page on Brightspace.
    - It is the student's responsibility to check the course page on a regular basis.
-

# Medical Absences/Sick Notes

---

- If you are absent from class as a direct result of personal illness or distress
    - Complete the “Student Declaration of Absence” form
    - Upload it to the Medical Absences dropbox on Brightspace within 3 calendar days following your absence
    - You are responsible for making up for the missed content
  - Note that the “Student Declaration of Absence”:
    - is valid for quizzes
    - is not valid for assignments or class project.
    - is not valid for the final examination
    - can only be submitted twice in the semester
-

# Makeup / Late Submission Policy

---

- There will be no makeup quizzes or tests. The following rules will be applied if you submit the “student declaration of absence” within 3 calendar days:
    - **For missed quizzes:** You will receive either the average of all your remaining quizzes, or the class average for that quiz, whichever is **lower**.
    - **For missed assignment/project deadlines:** Assignments can be started early to avoid problems later nearer the deadline. For unavoidable circumstances, please contact me BEFORE the deadline and an extension may be granted. Otherwise, late assignments will receive a zero mark.
  - No Email submissions of any kind are allowed
    - These include assignments, and declaration of absence.
-

# A Note on Plagiarism

---

- Plagiarism is defined as:
  - to steal and pass off (the ideas or words of another) as one's own
  - to use (another's production) without crediting the source
  - to commit literary theft
  - to present as new and original an idea or product derived from an existing source
- In other words, plagiarism is an act of fraud. It involves both stealing someone else's work and lying about it afterward

[Source: <http://www.plagiarism.org/plagiarism-101/what-is-plagiarism/>]

- Plagiarism of any sort will result in a report to AIO. Note that we have a zero tolerance policy towards plagiarism
  - Dal's Academic Integrity website has all the details you need (<http://academicintegrity.dal.ca.>)
  - *Note: For plagiarism detection, your submitted assignments may be uploaded to servers that are outside Canada. If you do not want this, please let me know in advance.*
-

# Other Policies

---

- **Responsible Computing Policy**

- Usage of all computing resources in the Faculty of Computer Science must be within the Dalhousie Acceptable Use Policies (<http://its.dal.ca/policies/>) and the Faculty of Computer Science Responsible Computing Policy. ([https://www.cs.dal.ca/downloads/fcs\\_policy\\_local.pdf](https://www.cs.dal.ca/downloads/fcs_policy_local.pdf))

- **Culture of Respect**

- Every person has a right to respect and safety. We believe inclusiveness is fundamental to education and learning. Misogyny and other disrespectful behaviour in our classrooms, on our campus, on social media, and in our community is unacceptable. As a community, we must stand for equality and hold ourselves to a higher standard. (<http://www.dal.ca/cultureofrespect.html>)

- **Grade Scale and Definitions:**

- The Senate approved grade scale is available at:
  - [http://www.dal.ca/campus\\_life/academic-support/grades-and-student-records/grade-scale-and-definitions.html](http://www.dal.ca/campus_life/academic-support/grades-and-student-records/grade-scale-and-definitions.html)
-

# Other Policies

---

- Academic Integrity
    - [http://www.dal.ca/dept/university\\_secretariat/academic-integrity.html](http://www.dal.ca/dept/university_secretariat/academic-integrity.html)
  - Accessibility
    - [http://www.dal.ca/campus\\_life/student\\_services/academic-support/accessibility.html](http://www.dal.ca/campus_life/student_services/academic-support/accessibility.html)
  - Student Code of Conduct
    - [https://www.dal.ca/campus\\_life/safety-respect/student-rights-and-responsibilities/student-life-policies/code-of-student-conduct.html](https://www.dal.ca/campus_life/safety-respect/student-rights-and-responsibilities/student-life-policies/code-of-student-conduct.html)
  - Recognition of Mikmaq Territory
    - Visit the office in the McCain Building (room 3037) or contact the programs at [elders@dal.ca](mailto:elders@dal.ca)
  - Learning and Support Resources
    - General Academic Support — Advising  
[http://www.dal.ca/campus\\_life/student\\_services/academic-support/advising.html](http://www.dal.ca/campus_life/student_services/academic-support/advising.html)
    - Fair Dealing Guidelines  
<https://libraries.dal.ca/services/copyright-office/guidelines/fair-dealing-guidelines.html>
    - Dalhousie University Library  
<http://libraries.dal.ca>
-



# Work Hard and Best of Luck

---

- Office location: 205, Goldberg CS building
  - Lectures: T R 08:35 – 9:55 (Kenneth Rowe 1009)
  - Lab: M 10:05-11:25 & 11:35-12:55  
(McCain Arts&SS 2019)
  - Office hours: Wed, 10:30 – 11:30  
(or send me an email)
  - E-mail: [kaziz@dal.ca](mailto:kaziz@dal.ca)
-

# Warm-up Exercises – 1

---

- Write a C++ program to calculate the Factorial of a number.

# A possible solution

---

```
#include <iostream>
int main()
{
    int n;
    long rslt = 1;
    std::cout << "Enter a number: ";
    std::cin >> n;
    for(int i = 1; i <=n; ++i)
    {
        rslt *= i;
    }
    std::cout << "Factorial of " << n << " = " << rslt;
    return 0;
}
```

---

# Another Solution

---

```
#include<iostream>
using namespace std;

class factorial{
    int rslt, n;
    public:
        void fac();
        void show();
};

void factorial::fac()
{
    rslt=1;
    cout<<"Enter a number: ";
    cin>>n;
    for(int i=1;i<=n;i++)
        rslt *= i;
}
```

```
void factorial::show()
{
    cout<<"\nFactorial of "<<n<<" is
"<<rslt<<endl;
}

int main()
{
    clrscr();
    factorial obj_f;
    obj_f.fac();
    obj_f.show();
    return 0;
}
```

---

# Warm-up Exercises – 2

---

```
#include <iostream>
using namespace std;

int main()
{
    int *i;
    int *j = new int;
    i = j;
    *j = 10;
    delete i;
    i = new int;
    *i = 62;
    j = new int;
    j = i;
    *j = 26;
    cout<<*i<<" "<<*j<<endl;
    return 0;
}
```

---