

# Object Oriented Programming Techniques

(CSE142)

Syllabus

Spring'26

## Contents

<b>1 Basic Information</b>	<b>2</b>
1.1 Objectives . . . . .	2
1.2 Knowledge and Skills . . . . .	2
<b>2 Course Contents</b>	<b>2</b>
2.1 Weekly Schedule (tentative) . . . . .	3
2.2 References . . . . .	3
<b>3 Logistics</b>	<b>3</b>
3.1 Staff and Office Hours . . . . .	3
<b>4 Administration</b>	<b>5</b>
4.1 Assessments . . . . .	5
4.2 Grading . . . . .	5
4.3 Project . . . . .	6
4.4 Late Work and Makeup Policy . . . . .	6
4.5 Attendance Policy . . . . .	7
4.6 Academic Integrity . . . . .	7
4.7 FAQs . . . . .	7
<b>5 Objectives and Outcomes</b>	<b>8</b>
5.1 Course Learning Outcomes . . . . .	8
5.2 Program Learning Outcomes/Graduate Attributes . . . . .	8
5.3 Assessments / CLO's / PLO's Mapping . . . . .	9

# 1 Basic Information

Credits	3 theory + 1 lab hours
Prerequisites	CSE-141 Introduction to Programming

## 1.1 Objectives

This course builds on the foundational knowledge from ‘CSE-141 Introduction to Programming’ and expands into more advanced topics in computer science and software development. The course introduces students to the object-oriented programming paradigm and its implementation in C++. It also covers the related functional and generic programming constructs prevailing in modern programming languages. The course also introduces students to the standard template library (STL) and its use in solving problems. This course prepares students for the more advanced programming courses such as ‘CSE-247 Data Structures’.

## 1.2 Knowledge and Skills

On completion of the course the student should have the following learning outcomes defined in terms of knowledge and skills:

- **Knowledge**

The student

- knows the basics of object-oriented, functional, and generic programming
- understands how the above programming styles may be used to solve problems

- **Skills**

The student is able to

- construct small programs, using the programming languages covered, that include the use of user-defined data types, inheritance, generic programming, and lambda expressions
- decompose problems into modular structures and express functionality in terms of abstract data type (ADT) or application programming interface (API)
- make use of available program libraries
- use integrated development environments, command-line tools, and a version control system

## 2 Course Contents

This course focuses on introducing students to advanced C++ programming concepts such as classes, objects, inheritance, exception handling, smart pointers, templates, linked lists, resizable arrays, stacks, queues, standard library functions, and lambda expressions. The project development allows students to apply their knowledge and skills learned throughout the course in a

practical context. The final week is dedicated to reviewing all topics covered in the course and preparing for the final exam.

## 2.1 Weekly Schedule (tentative)

## 2.2 References

Following are useful references:

- *Introduction to C++* by George Tselikis, CRC Press, 2023.
- *Modern C++ for Absolute Beginners: A Friendly Introduction to the C++ Programming Language and C++11 to C++23 Standards* by Slobodan Dmitrovic, Apress, 2nd Edition, 2023.
- *C++ Primer* by Lippman, Lajoie, and Moo, 5th Edition, Addison-Wesley, 2012.
- *Tutorials: C++ Language* at <https://cplusplus.com/doc/tutorial/>.

## 3 Logistics
















### 3.1 Staff and Office Hours

Table 2: Office Hours

Name	Role	Email	Office Location	Office Hours
Behraj Khan	Instructor	behrajkhan@iba.edu.pk	Room 205, Tabba Building, Main Campus	Tue/Thu: <b>11:00–13:00</b>
Samreen Kazi	Instructor	sakazi@iba.edu.pk	Faculty Lounge, City Campus	Tue/Thu: <b>3:30–5:30 pm</b>
Sadaf Alvi	Instructor	sadafalvi@iba.edu.pk	Faculty Lounge, City Campus	Tuesday: <b>8:30 am–12:30 pm</b>
Usman Ali	Instructor	uali@iba.edu.pk	BDA Lab (21-B), Tabba Academic Block, Main Campus	Tue: <b>10:30–12:45</b> Fri: <b>11:30–1:15</b>

Table 1: Weekly schedule of topics and assessments

<b>Week 1</b> <i>Jan 19 – Jan 24</i>	<b><u>Introduction to Classes and Objects</u></b> Class, instance variables, member functions (methods), get/set functions. Encapsulation, public private access specifiers, this keyword.	
<b>Week 2</b> <i>Jan 26 – Jan 31</i>	<b><u>Abstract Data Types</u></b> Hiding implementation details, class interface (API), constructors, default constructor, destructor, implementing operators, friend keyword.	
<b>Week 3</b> <i>Feb 2 -Feb 7</i>	<b><u>Stack and Queue ADTs</u></b> Implementing stack using resizable array, copy/move constructor.	
<b>Week 4</b> <i>Feb 9 – Feb 14</i>	<b><u>Stacks, Queues</u></b> resizable arrays, time-complexity	<i>HW Out 1</i>
<b>Week 5</b> <i>Feb 16 – Feb 21</i>	<b><u>Inheritance and Polymorphism</u></b> Deriving classes, protected access specifier, function overriding, virtual function.	<i>HW Due 1</i>
<b>Week 6</b> <i>Feb 23 –Feb 28</i>	<b><u>Abstract Class and Interface</u></b> Pure virtual functions, C++ abstract class, interface class, virtual destructor.	<i>HW Out 2</i>
<b>Week 7</b> <i>Mar 2 – Mar 7</i>	<b><u>Exception Handling</u></b> try - catch blocks, multiple catch blocks, custom exceptions.	<i>HW Due 2</i>
Mar 9 – Mar 14	<b>Midterm Examinations</b>	
Mar 16 – Mar 21	<b>Semester Break</b>	
<b>Week 8</b> <i>Mar 23 – Mar 28</i>	<b><u>Generic Programming</u></b> Generic functions, generic types, templates, auto type deduction, concepts.	<i>HW Out 3</i>
<b>Week 9</b> <i>Mar 30 – Apr 4</i>	<b><u>Functional Programming</u></b> Functions as arguments, function objects, function pointers,lambda expressions, capture list, map, filter, reduce.	
<b>Week 10</b> <i>Apr 6 – Apr 11</i>	<b><u>Elementary Sorts</u></b> Selection sort, insertion sort, time complexity.	<i>HW Due 3</i>
<b>Week 11</b> <i>Apr 13 – Apr 17</i>	<b><u>Recursion</u></b> Backtracking algorithms.	<i>HW Out 4</i>
<b>Week 12</b> <i>Apr 20 – Apr 25</i>	<b><u>Mergesort</u></b> Top-down mergesort, bottom-up mergesort, time complexity.	
<b>Week 13</b> <i>Apr 27 – May 2</i>	<b><u>Quicksort</u></b> Partitioning, quicksort, time complexity.	
<b>Week 14</b> <i>May 4 – May 9</i>	<b><u>Libraries</u></b>	<i>HW Due 4</i>
<b>Week 15</b> <i>May 11 – May 16</i>	<b><u>Review</u></b> Review of all topics covered in the course, preparation for final exam.	<i>Final Exam</i>

Course	Instructor	Location	Schedule
 Theory 100475	Sadaf Alvi	MTC-16 (North), Tabba Block	Mon/Wed: 08:30–09:45
 Lab 100475	Sadaf Alvi	CCL4, HBL Block, City Campus	Thu: 08:30–11:15
 Theory 100877	Sadaf Alvi	MTC-16 (North), Tabba Block	Mon/Wed: 10:00–11:15
 Lab 100877	Sadaf Alvi	CCL4, HBL Block, City Campus	Thu: 16:00–18:45
 Theory 100488	Usman Ali	MTC-18, Tabba Block	Mon/Wed: 16:00–17:15
 Lab 100488	Usman Ali	MCL-1, Aman CED	Fri: 08:30–11:15
 Theory 100481	Samreen Kazi	C5, Aman Tower, City Campus	Tue/Thu: 11:30–12:45
 Lab 100481	Samreen Kazi	CCL4, HBL Block, City Campus	Sat: 11:30–14:15
 Theory 100482	Samreen Kazi	C5, Aman Tower, City Campus	Tue/Thu: 13:00–14:15
 Lab 100482	Samreen Kazi	CCL4, HBL Block, City Campus	Sat: 14:30–17:15
 Theory 100485	Behraj Khan	MTC-17 (North), Tabba Block	Mon/Wed: 17:30–18:45
 Lab 100485	Behraj Khan	MCL-2 (111 Lab), Aman CED	Fri: 14:30–15:45
 Lab 100485	Behraj Khan	MCL-2 (111 Lab), Aman CED	Fri: 16:00–17:15
 Theory 100483	Behraj Khan	MTC-17 (North), Tabba Block	Mon/Wed: 16:00–17:15
 Lab 100483	Behraj Khan	MCL-2 (111 Lab), Aman CED	Sat: 08:30–11:15

## 4 Administration

### 4.1 Assessments

We have several assessment instruments designed for this course namely, homework, quizzes, project, midterm and final exams. All of these tools are important for learning the concepts we discuss in the class. The ‘personal struggle’ you engage in with these homework problems, quizzes, project, and exams will allow you to develop the skills necessary for success as a computer scientist. As a student in this class it is expected that you will always spend some time thinking about the problems/questions (on homework and project) on your own before asking for hints, looking up solutions etc. Do not go in search of complete solutions online; learning the material happens when you are working on problems rather than looking up complete solutions.

You are welcome to collaborate on homework, provided that (1) you write up your solutions *individually*, and (2) you clearly *cite the names* of all collaborators and sources. Failure to do so will result in **zero credit**. An additional key requirement is that you should be able to explain what you submit. Inability to do so will result again in zero credit.

### 4.2 Grading

A *custom/relative* grading scale will be used based on overall performance of the class and difficulty-level of the class assessments as judged by the instructors. The final grade will be based on the following components:

Assessment Component	Weight
Quizzes / Homework	15%
Project	10%
Lab exam	10%
Lab activities	15%
Midterm examination	20%
Final examination	30%

### 4.3 Project

An important component of this course is the project. It will be done in groups of 2-3 students. The students are expected to finalize their groups and project topic by the end of the third week of the course. It will be due in the last week of the course. The project will be graded based on the following criteria:

- **Design:** The design of the project should be modular and should follow the object-oriented design principles. The design should be well documented and should be easy to understand.
- **Implementation:** The project should be implemented in C++ and should be well documented. The code should be well structured and should follow the coding standards discussed in the class.
- **Testing:** The project should be well tested and should be able to handle all possible inputs. The project should be able to handle all possible errors and should be able to recover from them gracefully.
- **Presentation:** The project should be presented in the class in the last week of the course. The presentation should be well prepared and should cover all aspects of the project.
- **Report:** The project should be well documented and should be able to be compiled and run on any system. The report should be well written and should cover all aspects of the project.
- **Individual Contribution:** Each member of the group should be able to explain the project and should be able to answer questions related to the project.
- **Plagiarism:** The project should be original and should not be copied from any source. Any plagiarism will result in zero credit for the project.

### 4.4 Late Work and Makeup Policy

No late solutions will be accepted and no make-up for exams or any of the quizzes will be given. If you have a valid medical excuse (for any of the quizzes, homework, midterm exam, viva etc.), the percentage of your grade corresponding to the missed work will be shifted to the next exam. Valid excuses require supporting documentation from a doctor (standard IBA policy will prevail in all cases of conflicts).

## 4.5 Attendance Policy

IBA attendance policy applies.

## 4.6 Academic Integrity

Each student in this course is expected to abide by the IBA Code of Conduct. Scholastic dishonesty shall be considered a serious violation of these rules and regulations and is subject to strict disciplinary action as prescribed by IBA regulations and policies. Scholastic dishonesty includes, but is not limited to, cheating on exams, plagiarism on assignments, and collusion.

- **PLAGIARISM:** Plagiarism is the act of taking the work created by another person or entity and presenting it as one's own for the purpose of personal gain or of obtaining academic credit. Plagiarism includes the submission of or incorporation of the work of others without acknowledging its provenance or giving due credit according to established academic practices. This includes the submission of material that has been appropriated, bought, received as a gift, downloaded, or obtained by any other means. Students must not, unless they have been granted permission from all faculty members concerned, submit the same assignment or project for academic credit for different courses.
- **CHEATING:** The term cheating shall refer to the use of or obtaining of unauthorized information in order to obtain personal benefit or academic credit.
- **COLLUSION:** Collusion is the act of providing unauthorized assistance to one or more persons or of not taking the appropriate precautions against doing so. Any student violating academic integrity a second time in this course will receive a failing grade for the course, and additional disciplinary sanctions may be administered.

## 4.7 FAQs

- **Q.** What type of question will be asked in exam or quiz?  
**A.** See previous exams / quizzes on LMS.
- **Q.** Would you do relative grading?  
**A.** A custom grading scale will be used based on my assessment of exam difficulty level and overall class performance.
- **Q.** I have missed the quiz because of xyz reason.  
**A.** Best  $n - 1$  quizzes would be considered in the final grade.
- **Q.** Syllabus for midterm, final and lab exams?  
**A.** Everything discussed in class / practice problems / homeworks.

## 5 Objectives and Outcomes

### 5.1 Course Learning Outcomes

<i>Course Learning Outcomes (CLO's)</i>	
<b>CLO-1</b>	Acquire knowledge of underlying concepts of the object-oriented paradigm.
<b>CLO-2</b>	Design and implement object-oriented solutions for small systems involving single/multiple objects.
<b>CLO-3</b>	Students will learn team management and work distribution.

### 5.2 Program Learning Outcomes/Graduate Attributes

Graduate attributes (program learning outcomes - PLO's) taken from

<https://www.seoulaccord.org/document.php?id=79>.

- **PLO-1. Academic Education**

*[Educational depth and breadth]*

Completion of an accredited program of study designed to prepare graduates as computing professionals

- **PLO-2. Knowledge for Solving Computing Problems**

*[Breadth and depth of education and type of knowledge, both theoretical and practical]*

Apply knowledge of computing fundamentals, knowledge of a computing specialization, and mathematics, science, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements

- **PLO-3. Problem Analysis**

*[Complexity of analysis]*

Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines

- **PLO-4. Design / Development of Solutions**

*[Breadth and uniqueness of computing problems, i.e., the extent to which problems are original and to which solutions have previously been identified or codified]*

Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations

- **PLO-5. Modern Tool Usage**

*[Level and appropriateness of the tool to the type of activities performed]*

Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations

- **PLO-6. Individual and Team Work**

*[Role in, and diversity of, the team]*

Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary settings

- **PLO-7. Communication**

*[Level of communication according to type of activities performed]*

Communicate effectively with the computing community and with society at large about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions

- **PLO-8. Computing Professionalism and Society**

*[No differentiation in this characteristic except level of practice]*

Understand and assess societal, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice

- **PLO-9. Ethics**

*[No differentiation in this characteristic except level of practice]*

Understand and commit to professional ethics, responsibilities, and norms of professional computing practice

- **PLO-10. Life-long Learning**

*[No differentiation in this characteristic except level of practice]*

Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional

### 5.3 Assessments / CLO's / PLO's Mapping

The following table describes the mapping of course learning outcomes (CLO's) to program learning outcomes (PLO's).

The following table identifies the assessment instruments will be used to assess the CLO's.

	Academic Education	Knowledge for Solving Computing Problems	Problem Analysis	Design / Development of Solutions	Modern Tool Usage	Individual and Team Work	Communication	Computing Professionalism and Society	Ethics	Life-long Learning
	PLO-1	PLO-2	PLO-3	PLO-4	PLO-5	PLO-6	PLO-7	PLO-8	PLO-9	PLO-10
CLO-1		✓								
CLO-2				✓						
CLO-3										

Table 3: Mapping of CLO's to PLO's

	CLO-1	CLO-2	CLO-3
	Knowledge of OOP concepts	Design implement OOP solution	Team work
Labs		✓	✓
Homeworks	✓		
Quizzes	✓	✓	
Exams	✓	✓	
Project		✓	✓

Table 4: Mapping of assessment instruments to CLO's