National University of Sciences & Technology (NUST)
School of Electrical Engineering and Computer Science (SEECS)
Department of Computer Science

# Object Oriented Programming

| | | | |
|---|---|---|---|
| **Course Code:** | CS 212 | **Semester:** | 2nd |
| **Credit Hours:** | 3+1 | **Prerequisite Codes:** | CS-110: Fundamentals of Computer Programming |
| **Instructor:** | Shamyl Bin Mansoor | **Class:** | BESE-5 AB |
| **Office:** | Room A-308, SEECS | **Telephone:** | 051-9085-2173 |
| **Lecture Days:** | Monday, Tuesday, Thursday | **E-mail:** | shamyl.mansoor@seecs.edu.pk |
| **Class Room:** | | **Consulting Hours:** | Mon & Wed 4-5 (Sec B)<br>Mon 4-5 & Wed 10-11 (Sec A) |
| **Lab Engineer:** | Akhtar Munir | **Lab Engineer Email:** | akhtar.munir@seecs.edu.pk |
| **Knowledge Group:** | | **Updates on LMS:** | Beginning of the Week |

## Course Description:

The C++ language is used. The language constructs discussed include classes, inheritance, encapsulation, polymorphism, class derivation, abstract classes, static class members, object construction and destruction, namespaces, exception handling, function overloading and overriding, function name overload resolution, container classes, template classes, etc.

## Course Objectives:

The objective of this course is to understand the object oriented programming paradigm. The goal is to enable the students to model their problems in the Object Oriented Programming domain. In this course this is done using C++ as the programming language, although for the End Semester Project use of any other OOP language is encouraged.

| Course Learning Outcomes (CLOs): | | |
|---|---|---|
| **At the end of the course the students will:** | **PLO** | **BT Level*** |
| 1. Describe the Object Oriented Programming concept and be able to discuss the differences between procedural and object oriented languages | 1,6,10 | C-1, C-2 |
| 2. Demonstrate the ability to create and use classes within the C++ programming language | 2,3,4,5 | C-4, C-5, C-6 |
| 3. Develop programs using important C++ techniques, such as composition of objects, operator overloads, dynamic memory allocation, inheritance and polymorphism, file I/O, exception handling, templates, preprocessor directives, and basic data structure | 3,5 | C-1, C-2 |

| | | |
|---|---|---|
| 4. Build C++ classes using appropriate encapsulation and build-in classes | 1,2,3,5 | C-1, C-2 |
| 5. Apply the OOP concepts to solve complex problems independently | 1,2,3,4,11 | C-4, C-5, C-6 |
| * BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain | | |

## Mapping of CLOs to Assessment Modules and Weightages

| Assessments/CLOs | CLO1 | CLO2 | CLO3 | CLO4 | CLO5 |
|---|---|---|---|---|---|
| **Theory (75%)** | | | | | |
| **Quizzes:** | | | | | **15%** |
| **Assignments:** | | | | | **10%** |
| **OHT-1:** | | | | | **15%** |
| **OHT-2:** | | | | | **15%** |
| **End Semester Exam:** | | | | | **45%** |
| **Practical (25%)** | | | | | |
| **Labs: Lab practical** | | | | | **70%** |
| **Project:** | | | | | **30%** |

## Books:

**Text Book:**

1. C++ How To Program, Deitel& Deitel, Publication Date: March 25, 2011 | ISBN-10: 0132662361 | ISBN-13: 978-0132662369 | Edition: 8

**Reference Books:**

1. Ivor Horton's Beginning ANSI C++: The Complete Language, Third Edition Publication Date: January 8, 2004, ASIN: B0042NGRS0
2. Object Oriented Programming in C++ by Robert Lafore, Fourth Edition, Publication Date: December 19, 2001 | ISBN-10: 0672323087 | ISBN-13: 978-0672323089 | Edition: 4
3. Problem solving abstraction and design using C++ by F.L. Friedman. Addison Wesley, Fifth Edition, Publication Date: July 24, 2006 | ISBN-10: 0321433327 | ISBN-13: 978-0321433329 | Edition: 5
4. Thinking in C++ (B. Eckel), Publication Date: March 15, 2000 | ISBN-10: 0139798099 | ISBN-13: 978-0139798092 | Edition: 2

## Topics to be Covered:

| | |
|---|---|
| Object oriented concepts | Inheritance |
| Constructor & destructor | Multiple Inheritance |
| Setter & getter functions | Diamond problem |
| this pointer & copy constructor | Polymorphism |
| Friend functions | Abstract classes |
| Friend classes | Composition |
| Static class members | Aggregation |
| Static objects | Templates |
| Constant class members | Sequential file handling |
| Constant objects | Random file access |
| Operator Overloading | Exception handling |

| | Topic |
|---|---|
| **Week 1** | Lecture-1: C Fundamentals<br>Lecture-2: Revision Pointers<br>Lecture-3 Structures |
| **Week 2** | Lecture-4: Structures continue<br>Lecture-5: Nested structures<br>Lecture-6: Object oriented concept |
| **Week 3** | Lecture-7: Introduction to classes<br>Lecture-8: Need of Constructor and its usage<br>Lecture-9: Need of Destructor and its usage |
| **Week 4** | Lecture-10: Inline Member Functions<br>Lecture-11: Getter and setter functions<br>Lecture-12: this pointer |
| **Week 5** | Lecture-13: Copy Constructor<br>Lecture-14 &15 revision |
| **Week 6** | **OHT-1** |
| **Week 7** | Lecture-16:Friend Functions<br>Lecture-17: Friend Classes<br>Lecture-18: Static Class Members |

| | |
|---|---|
| **Week 8** | Lecture-19: Operator Overloading –i<br>Lecture-20: Operator Overloading –ii<br>Lecture-21: Introduction to Inheritance |
| **Week 9** | Lecture-22 & 23: Inheritance<br>Lecture-24 Case study – Inheritance |
| **Week 10** | Lecture-25: Virtual Functions<br>Lecture-26: Polymorphism<br>Lecture-27: dynamic conversion |
| **Week 11** | Lecture-28 & 29 Case study – Polymorphism<br>Lecture-30: Revision |
| **Week 12** | **OHT-** 2 |
| **Week 13** | Lecture-31: Abstract Classes<br>Lecture-32: Virtual Destructors<br>Lecture-33: Exception handling |
| **Week 14** | Lecture-34: Composition<br>Lecture-35: Case study – Composition<br>Lecture-36: Aggregation |
| **Week 15** | Lecture-37: Case study – Aggregation<br>Lecture-38: Templates – Function<br>Lecture-39: Templates - classes |
| **Week 16** | Lecture-40: Standard Template Library – Containers, Algorithms, Iterators<br>Lecture-41: Sequential Containers & Iterators<br>Lecture-42: Specialized Iterators and Associative Containers |
| **Week 18** | **ESE** |

| Lab Experiments |
|---|
| Lab 1: A Review of Structured Programming & Introduction to Classes |
| Lab 2: Classes and Constructors |
| Lab 3: Constructors and Destructors |

| |
|---|
| Lab 4: Const. Objects and Member Variables |
| Lab 5: Classes, Constructors |
| Lab 6: Aggregation and Inheritance |
| Lab 7: Inheritance and Operator Overloading |
| Lab 8: Polymorphism |
| Lab 9: File handling (part1) |
| Lab 10: File handling (part2) |
| Lab 11: Exception Handling (part1) |
| Lab 12: Exception Handling (part2) |

| Grading Policy: | |
|---|---|
| **Quiz Policy:** | The quizzes will be unannounced and normally last for ten minutes. The question framed is to test the concepts involved in last few lectures. Number of quizzes that will be used for evaluation is at the instructor's discretion. Grading for quizzes will be on a fixed scale of 0 to 10. A score of 10 indicates an exceptional attempt towards the answer and a score of 1 indicates your answer is entirely wrong but you made a reasonable effort towards the solution. Scores in between indicate very good (8-9), good (6-7), satisfactory (4-5), and poor (2-3) attempt. Failure to make a reasonable effort to answer a question scores a 0. |
| **Assignment Policy:** | In order to develop comprehensive understanding of the subject, assignments will be given. Late assignments will not be accepted / graded. All assignments will count towards the total (No 'best-of' policy). The students are advised to do the assignment themselves. Copying of assignments is highly discouraged and violations will be dealt with severely by referring any occurrences to the disciplinary committee. The questions in the assignment are meant to be challenging to give students confidence and extensive knowledge about the subject matter and enable them to prepare for the exams. |
| **Lab Conduct:** | The labs will be conducted for three hours every week. The students are to submit their results by submitting their lab tasks at the end of lab for evaluation. Students will be evaluated using online LMS quizzes. |
| **Plagiarism:** | SEECS maintains a zero tolerance policy towards plagiarism. While collaboration in this course is highly encouraged, you must ensure that you do not claim other people's work/ ideas as your own. Plagiarism occurs when the words, ideas, assertions, theories, figures, images, programming codes of others are presented as your own work. You must cite and acknowledge all sources of information in your assignments. Failing to comply with the SEECS plagiarism policy will lead to strict penalties including zero marks in assignments and referral to the academic coordination office for disciplinary action. |

| Tools / Software Requirement: | |
|---|---|
| | MS Visual Studio 2008/2010/2012 |