

CS217 Object Oriented Programming

Spring 2019 - NUCES, Peshawar Campus

Course Code:	CS217
Course Title:	Object Oriented Programming
Batch:	BS CS18
Total Credits:	3
Instructor:	Dr. Nauman
Contact:	(091) 111-128-128 (Ext 110)
Grade Distribution:	Assignments (10%) Quizzes (10%) Project (20%) Midterms (20%) Final Exam (40%)

1 Objectives

This course is a continuation of the Programming Fundamentals course of the first semester. This is also a core course and as such covers some fundamental concepts related to programming.

The primary objective of this course is to cover concepts such as Object Oriented Programming and the details of how the source code written in a mid-level language is converted to machine code. Intricate details of the workings of programs will be covered.

Details of the OO-paradigm will be discussed along with their practical implications. The programming language of choice will be C++. By the end of the course, students should be comfortable with the syntax of C++ as well as its application to the OO-paradigm. They should be able to work with large scale projects and break down a program into smaller, more manageable chunks.

2 Course Books

1. Herbert Schildt. **C++ The Complete Reference**. Fourth Edition. McGraw-Hill, 2012.
2. Bruce Eckel. **Thinking in C++: Introduction to Standard C++**. Volume 1. Second Edition. Pearson. 2013.

3 Reference Materials

- Abelson, Harold, Gerald Jay Sussman, and Julie Sussman. Structure and interpretation of computer programs. Justin Kelly, 1996.
- Knuth, Donald. The Art of Programming. Addison-Wesley. 1968.
- Bruce Eckel. **Thinking in C++: Standard Libraries and Advanced Topics**. Volume 2. Second Edition. Pearson. 2013.

4 Course Policies

Attendance:	At least 80% attendance is required by each student. Students failing to have the required attendance will not be allowed to appear in the final exam resulting in 'F' grade in the course. No relaxation shall be given in this regard.
Quizzes:	Quizzes will be closed book. Most quizzes will be announced a day or two in advance but unannounced quizzes are also possible.

Semester Project:	Each student will be required to create, document and present a semester project. Details of the project will be provided in a separate document.
Marks Contest Deadlines:	The marks contest deadlines for all instruments is 3 days after marks are announced. NO changes will be entertained after that.
Plagiarism Policy:	Copying, cheating or any other academic dishonesty will be penalized with zero marks in that category. This applies much more rigidly to the semester project.
Reading Tasks:	Reading tasks will be given each week. Presentations will also be required from the students on topic of relevance from time to time.
Mode of presentation:	Methods of presentation will be lectures, demonstrations and practical sessions.

5 Topic Breakdown

This breakdown is subject to minor changes in topics during the semester.

Week	Topics
Week 1	Course orientation, data types, compilation, cin/cout, functions and prototypes Decisions, iteration
Week 2	Arrays, 2D arrays, char arrays, strings, struct basics Pointers (vars, arrays), passing by value versus reference
Week 3	Pointers to structures, basic list with structs File handling – reading and writing files
Week 4	File handling - binary files Header files, code structure, stack and heaps, breaking down code into multiple files
Week 5	Code structure in RAM, code, data segments etc., compiler, linker and loader Macros and pre-processing directives, include libraries, const and inline
Week 6	Sessional I Pointers to functions and callbacks
Week 7	Intro to Classes and Objects, attributes, methods Constructor/destructor, enums
Week 8	Inheritance and composition, static keyword Namespaces
Week 9	Public, private, protected, friend - encapsulation, data hiding Class templates
Week 10	Virtual functions, polymorphism Example 1: ADT that supports push and pop, stack and queue (with arrays)
Week 11	Example 2: list with class pointers Sessional II
Week 12	Standard Template Library - containers STL - iterators
Week 13	STL - algorithms - searching STL - algorithms - sorting
Week 14	Bitwise operations, and, or, not, shift, flags Boost library
Week 15	SFML - I SFML - II
Week 16	Project discussion