

COURSE DESCRIPTION

Department and Course Number	CS 342	Course Coordinator	Behzad Parviz
Course Title	Object-Oriented Programming Using C++	Total Credits	4

Current Catalog Description:

Concepts and techniques of Object Oriented Programming; structured design and modular construction; use of C to demonstrate structured programming; use of C++ to demonstrate fundamentals of Object Oriented Programming.

Textbook:

Deitel & Deitel. “C++ *How to Program*”, *Fourth Edition*, Prentice Hall, 2005.

Deitel & Deitel. “C++ *in the Lab*”, *Fourth Edition*, Prentice Hall, 2005.

References:

Savitch., “*Problem Solving with C++: The Object of Programming*”, *4th Ed.*, Addison Wesley, 20003.

Course Goals:

At the end of the course, students are able to

- Create C++ classes
- Implement encapsulation and data hiding
- Use inheritance and polymorphism
- Use stream I/O
- Handle C++ exceptions
- Create, read, and write files

This course is offered as a service course for non CS majors.

Prerequisites by Topic:

Introduction to Programming.

Major Topics Covered in the Course:

1. Classes and Objects
 - Structure Definition
 - Interface and Implementation
 - Constructors and Destructors
 - Set and get Functions
 - const Objects and const Member Functions

- friend Functions and friend Classes
- static Class Members
- Container Classes and Iterators

2. Operator Overloading; String and Array Objects

- Fundamentals of operator overloading
- Restrictions of operator overloading
- Overloading Stream-Insertion and Stream-Extraction Operators
- Overloading Unary Operators
- Overloading Binary Operators

3. Inheritance

- Base Classes and Derived Classes
- protected Members
- public, protected and private Inheritance
- Inheritance Hierarchy

4. Polymorphism

- Relationships Among Objects in a Inheritance Hierarchy
- Invoking Base-class Functions from Derived Class Objects
- Aiming Derived-Class Pointers at Base Class Objects
- Derived-Class Member-Function Calls via Base-Class Pointers
- Virtual Functions

5. Templates

- Function Templates
- Overloading Function Templates
- Class Templates
- Templates and Inheritance
- Templates and Friends
- Templates and static Members

6. C++ Stream Input/Output

- Streams
- Stream Output
- Stream Input
- Unformatted I/O using read, write, and count
- Introduction to Stream Manipulators
- Stream Format States and Stream Manipulators

7. Exception handling

- Overview
- Other Error-Handling Techniques
- Rethrowing an Exception
- Exception Specifications
- Process Unexpected Exceptions

Stack Unwinding
 Constructors, Destructors, and Exception Handling
 Exception and Inheritance
 Standard Library Exception Hierarchy

8. File Processing

Files and Stream
 Creating a Sequential-Access File
 Reading Data from a Sequential-Access File
 Random-Access File
 Creating a Random-Access File
 Reading data sequentially form a Random-Access File

Laboratory Projects (specify number of weeks on each):

Each week the students complete a 3-hour lab project on a selected topic. In addition, they design and develop three to five large projects covering the materials taught in the course.

- Week 1: Design a program using classes
- Week 2: Design a program using operator overloading
- Week 3: Design a program using string and array objects
- Week 4: Practice with classes and inheritance
- Week 5: Build a program using polymorphism. MIDTERM 1
- Week 6: Create a program using templates
- Week 7: Design a program using stream I/O.
- Week 8: Practice with C++ exception handling. MIDTERM 2
- Week 9: Design a program using files for I/O
- Week 10: Design a project using main topics covered in the course

Estimate Curriculum Category Content (Quarter Hours)

Area	Service*	Area	Service*
Algorithms	0.75	Data Structures	0.75
Software Design	1.0	Prog. Languages	1.5
Comp. Arch.		Other	

Service* : This course is offered as a service course for non CS majors.

Oral and Written Communications:

Written documentation of software built in labs and homework assignments.

Social and Ethical Issues:

No significant component.

Theoretical Content:

- Inheritance (1 week)
- Polymorphism (1 weeks)

Problem Analysis:

In the first part of the course, students learn the basic concepts of object-oriented programming, such as classes, inheritance, and polymorphism. In the second part of the course, they learn how to use templates, stream I/O, exception handling, and file processing.

Solution Design:

Solution design in this course mostly involves choosing appropriate OO techniques to perform programming tasks. It also involves generating pseudo-code for program development. Students learn how good program design reduces coding and debugging time.