

**SAN DIEGO COMMUNITY COLLEGE DISTRICT  
CITY, AND MIRAMAR COLLEGES  
ASSOCIATE DEGREE COURSE OUTLINE**

**SECTION I****SUBJECT AREA AND COURSE NUMBER:** Computer and Information Sciences 205**COURSE TITLE:** Object Oriented Programming using C++**Units: 4**  
Grade Only**CATALOG COURSE DESCRIPTION:**

This course introduces students to Object Oriented Programming (OOP) using the C++ programming language and includes the essential concepts related to OOP including use of classes and objects, inheritance, templates, polymorphism, pointers and references, and I/O streams. Students may apply this course to an Associate Degree or Certificate and may be transferred to CSU and private colleges and universities.

**REQUISITES:**

NONE

**FIELD TRIP REQUIREMENTS:** May be required**TRANSFER APPLICABILITY:** Associate Degree Credit & transfer to CSU and/or private colleges and universities UC Transfer Course List**TOTAL LECTURE HOURS:** 48 - 54**TOTAL LAB HOURS:** 48 - 54**STUDENT LEARNING OBJECTIVES:**

Upon successful completion of the course the student will be able to:

1. Use design tools such as hierarchy charts, class (Booch) diagrams, etc. to create an Object Oriented Design.
2. Construct classes containing both data members and member functions (methods) and instantiate objects of the classes.
3. Describe the relationship(s) between classes and objects.
4. Create both base classes and derived classes demonstrating both single and multiple-level inheritance.
5. Design and implement class templates.
6. Implement programming that demonstrates function overloading, method over-riding, and polymorphism.
7. Create programming that uses class interaction.
8. Use exceptions-handling utilities to control error conditions.

**SECTION II****1. COURSE OUTLINE AND SCOPE:****A. Outline Of Topics:**

The following topics are included in the framework of the course but are not intended as limits on content. The order of presentation and relative emphasis will vary with each instructor.

- I. Object oriented design fundamentals overview
  - A. Planning
  - B. Project Analysis
  - C. Program design
  - D. Code implementation
- II. A review of C++ programming fundamentals
  - A. Input and output operations
  - B. Variables
  - C. Arithmetic operators
  - D. Loops and decisions
- III. The structure and syntax of classes
  - A. Syntax diagram for classes
  - B. Constructor and destructor member functions (methods)
  - C. Syntax diagram for member functions
  - D. Global resolution operator
  - E. The private and public statements
  - F. Method and operator overloading
- IV. Class inheritance and method over-riding
  - A. Base class and derived class syntax
  - B. The friend statement
  - C. Multilevel inheritance
  - D. Inheritance of multiple base classes
  - E. Method over-riding
  - F. Use of this and super operators
- V. Function and class templates
  - A. Template syntax
  - B. Use of
  - C. Abstract methods
- VI. Virtual functions and polymorphism
  - A. Syntax of virtual methods and classes
  - B. Use of virtual
  - C. Use of references in creating virtual functions
  - D. Pure virtual methods and classes
  - E. Creation of polymorphic functions
  - F. Dynamic linkages
  - G. Virtual destructors
- VII. Pointers and references
  - A. Use of \*, &, and -> operators
  - B. Differences between references and pointers
- VIII. Exceptions handling
  - A. Use of try and catch
  - B. Use of throw
  - C. Creation of catch classes
  - D. Naming of exceptions
  - E. Handling uncaught exceptions
- IX. I/O streams
  - A. Keyboard and screen I/O
  - B. File I/O including ifstream, ofstream, and fstream classes
  - C. Use of I/O stream methods including binary and text file operations
  - D. Logical device and physical device access
  - E. Creation of I/O manipulators and using stream references
- X. The string stream
  - A. Use of istrstream, ostrstream, and strstream classes
  - B. Relationships between the various stream classes
  - C. Creating dynamic string streams

## **B. Reading Assignments:**

Reading assignments are required and may include but, are not limited to, the following:

**I. Books:**

- A. Jaeschke, R. The Dictionary of Standard C. Saddle River, New Jersey, Prentice Hall, Upper, 2001
- B. Strastrup, B. The C++ Programming Language. Menlo Park, California, Addison Wesley Publishing Company, 1991.

**II. Journals:**

- A. Journal of Object Oriented Programming. SIGS Publications, New York, New York.
- B. Object Magazine. New York, New York, SIGS Publications.
- C. The C++ Journal. Port Washington, New York.

**III. Internet Sites:**

- A. [www.cyberdiem.com/vin/learn.html](http://www.cyberdiem.com/vin/learn.html)
- B. [www.cygnus.com/misc/wp/](http://www.cygnus.com/misc/wp/)
- C. [www.research.att.com/~bs/C++.html](http://www.research.att.com/~bs/C++.html)
- D. [www.webnz.com/robert/spp\\_site.html](http://www.webnz.com/robert/spp_site.html)
- E. [www.cs.duke.edu/~ola/cplus.html](http://www.cs.duke.edu/~ola/cplus.html)

**C. Appropriate Assignments that Demonstrate Critical Thinking:**

Critical thinking assignments are required and may include, but are not limited to, the following:

- I. Analyze and evaluate programming utilities using object-oriented design.
- II. Create C++ libraries that use object oriented programming.
- III. Assess programming sequences to deductively and logically debug and correct error sequences.
- IV. Review current journal articles for algorithm concepts and programming utilities.

**D. Appropriate Outside Assignments:**

Outside assignments may include, but are not limited to, the following:

- I. Evaluate programming utilities used in standard programming sequences.
- II. Prepare interactive and batch processing classes and utilities.
- III. Create program evaluation designs to test student-developed programs.
- IV. Research the use of classes, inheritance and virtual programming techniques in industry.

**E. Writing Assignments:**

Writing assignments are required and may include, but are not limited to, the following:

- I. Description of the syntax and the use of C++ control structures including loops and decisions.
- II. Creation and use of standard design charts for designing classes.
- III. Creation of a program demonstrating the use of I/O streams and string streams.
- IV. Implementation of a program to demonstrate class inheritance.
- V. The writing of a sample program that demonstrates the use of polymorphic methods.
- VI. The implementation of exception handling in a program to control possible error conditions.
- VII. Critiquing prepared programming sequences and program packages for error analysis.

**2. METHODS OF EVALUATION:**

A student's grade will be based on multiple measures of performance unless the course requires no grade. Multiple measures may include, but are not limited to, the following:

- I. Performance on hands-on assignments.
- II. Written responses to in-class assignments.
- III. Responses to in-class objective and/or essay question quizzes and/or examinations.
- IV. Development of C++ programs using OOP techniques.
- V. Interactive one-on-one demonstration of program testing and operations.
- VI. Participation in class discussions.
- VII. Development of program documentation.

**3. METHODS OF INSTRUCTION:**

Methods of instruction may include, but are not limited to, the following:

- \* Distance Education
- \* Lecture
- \* Other (Specify)
  - \* 1. In-class practice of concepts and techniques included in the course objectives.
  - \* 2. Interactive group activities including analysis, evaluation, and modification of current applications.
  - \* 3. Distance learning students will attend electronic conferences, or where feasible, attend scheduled on-site conferences. Communication between distance learning students and the instructor will take place at least as specified in the course syllabus of the course with a minimum of two electronic communications during the semester between the student and the instructor.

#### **4. REQUIRED TEXTS AND SUPPLIES:**

Textbooks may include, but are not limited to:

##### **TEXTBOOKS:**

1. Jaeschke, R.. The Dictionary of Standard C, none ed. Prentice hall, 2001,
2. Strastrup B.. The C++ Programming Language, None ed. Addison Wesley, 1991,

##### **MANUALS:**

##### **PERIODICALS:**

##### **SOFTWARE:**

##### **SUPPLIES:**

**ORIGINATOR:** Rose LaMuraglia

**CO-CONTRIBUTOR(S)**

**DATE:** 10/13/2005