

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

**SECTION I****SUBJECT AREA AND COURSE NUMBER:** Computer and Information Sciences 192**COURSE TITLE:** C/C++ Programming**Units: 4**  
Grade Only**CATALOG COURSE DESCRIPTION:**

This course presents basic programming concepts using the C++ programming language. The organization of standard I/O classes is emphasized. Structured and Object oriented programming techniques are presented and used to design and implement a variety of programming problems.

**REQUISITES:****Advisory:**

ENGL 049 with a grade of "C" or better, or equivalent or Assessment Skill Level W5  
&  
CISC 186 with a grade of "C" or better, or equivalent

**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. Define data types, constants, and arithmetic operations.
2. Use variables and objects including declaration and instantiation, data assignment, and the use of cin and cout.
3. Use relational and logical operators and create Boolean expressions.
4. Use decision structures including if - else and switch.
5. Apply iteration structures including while, for, and do - while.
6. Create user - defined functions including parameter lists and returning values.
7. Create arrays, data assignments, data manipulation, and output from arrays.
8. Use arrays as parameters in user-defined functions.
9. Create and manipulate strings including character access and string parsing.
10. Use standard string manipulation functions including strcmp(), strcpy(), and strlen().
11. Employ pointer variables and describe the relationship between pointers, standard variables and arrays.
12. Create structures, arrays of structures, and a signally linked list of structures.
13. Use data files including writing and reading to and from both text and binary files.
14. Employ searching and sorting arrays.

## 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. Distance learning students will complete the same course content. The following topics may be included in the framework of the course but are not intended as limits on the content. The order of presentation and relative emphasis may vary with the instructor.

- A. Data types, Constants, and Arithmetic Operations.
  - 1. Algorithms and algorithm development.
  - 2. The form of a C++ function.
  - 3. Cin and cout as examples of C++ objects.
  - 4. Basic data types including char, int and double.
- B. Variables and objects.
  - 1. The declaration of variables and instantiation of objects.
  - 2. Integer qualifiers and data type conversions.
  - 3. Assignment statements.
  - 4. Accumulators and counters.
  - 5. Formatting data output.
- C. Relational and Logical Operators.
  - 1. Logical Operators: &&, ||, and !.
  - 2. If - else and switch structures and how they might be used in a programming sequence.
  - 3. Relational operators: >, =,
- D. Decision structures.
  - 1. Nested if - else and switch structures.
  - 2. Compound statements.
  - 3. Break and continue.
  - 4. Pre vs. post test loops.
- E. Iteration Structures.
  - 1. While(), for(), and do-while() loops.
  - 2. Nested loops.
  - 3. Using loops to check validity of data entered into interactive programming.
  - 4. Function and argument declarations.
  - 5. Function prototypes, functions headers and function calls.
- F. User-Defined Functions.
  - 1. Functions and the placement of statements.
  - 2. Function stubs and functions with empty parameter lists.
- G. Arrays, Data Assignments, Data Manipulations and Array Output.
  - 1. Declaration of arrays and data assignments.
  - 2. Data access in arrays and using both traverse techniques and random access techniques.
  - 3. Using arrays for searching and sorting data.
- H. Arrays as Parameters.
  - 1. Structures and arrays of structures.
  - 2. Structure types as function parameters.
  - 3. Strings, arrays of strings, and implement structures incorporating strings as data members.
- I. Strings.
  - 1. Character members of a string.
  - 2. The string.h library functions.
  - 3. Class strings.
- J. Variable pointers.
  - 1. Simple variable pointers.
  - 2. Arrays (array names) and pointers to arrays.
- K. Structures.

1. Pointers to structures.
  2. Pointer arithmetic.
  3. Dynamic Memory Allocations.
  4. I/O file systems.
- L. Data Files.
1. Text and binary data files.
  2. Data files and other devices.
  3. Random file access techniques.
  4. File streams as functions arguments.

**B. Reading Assignments:**

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

- I. 1. Bronson, G.J., Program Development and Design Using C++. PWS Publishing Co.: Boston, MA. 1997
- II. 2. Dale, et. al. Programming and Problem Solving with C++. Jones and Bartlett: Boston. 1997
- III. 3. Deitel, H.M. and P.J. Deitel. How to Program C++. Prentice Hall: Englewood Cliffs, N.J. 1994
- IV. 4. Freedman, F.L. and E.B. Koffman. Problem Solving, Abstraction, and Design Using C++. Addison Wesley: Reading, MA. 1994
- V. 5. Mercer, R. Computing Fundamentals with C++: Object Oriented Programming and Design. Franklin, Beedle and Associates: Watsonville, OR. 1999

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

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

- I. 1. Analyze and critique current programming packages.
- II. 2. Create program design sequences using a variety of design tools including design charts, HIPO charts screen design charts, and pseudocode.
- III. 3. Create and implement program designs using C++.
- IV. 4. Implement a variety of techniques to debug and correct error sequences.

**D. Appropriate Outside Assignments:**

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

- I. 1. Create interactive C++ programs.
- II. 2. Evaluate various development tools including HIPO charts, screen design techniques and the use of CASE tools.
- III. 3. Create program testing designs to validate the operation of programs.
- IV. 4. Describe the uses of C++ in the development of Web applications, creation of system-level utilities as well as stand alone applications.
- V. 5. Critique prepared programming sequences and program packages.

**E. Writing Assignments:**

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

- I. 1. The creation of programming documentation including design and implementation plans.
- II. 2. The creation of user instructions.
- III. 3. The preparation of documentation sequences, internal to the program code.

**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. A final grade of "C" or better indicates the student has the ability to successfully apply theory and techniques taught in this course and in subsequent courses and in practice. Distance learning students may submit their evaluation items electronically and receive electronic feedback.
- II. Performance on hands-on assignments. Written responses to in-class assignments. Responses to

in-class objective and/or essay question quizzes and/or examinations. Development of C++ programs. Interactive one-on-one demonstration of program testing and operations. Participation in class discussions. Development of program documentation.

### **3. METHODS OF INSTRUCTION:**

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

- \* Other (Specify)
- \* Distance Education
- \* 1. Computer assisted instruction.
- \* 2. Lecture.
- \* 3. Audio/visual aided instruction.
- \* 4. In-class practice of concepts and techniques included in the course objectives.
- \* 5. Interactive group activities including analysis, evaluation, and modification of current applications.
- \* 6. 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 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. Oualline, Steve. Practical C++ Programming, NA ed. O'Reilly & Associates, Inc., 1998, ISBN: NA
2. Stroustrup, Bjarne. The C++ Programming Language, 3rd ed. Addison Wesley Longman, Inc., Wesley, MA., 1997, ISBN: NA

#### **MANUALS:**

#### **PERIODICALS:**

#### **SOFTWARE:**

#### **SUPPLIES:**

**ORIGINATOR:** Ed Brunjes

**CO-CONTRIBUTOR(S)**

**DATE:** 06/03/2001