GEORGIA PERIMETER COLLEGE
MATHEMATICS ACADEMIC GROUP
COMMON COURSE OUTLINE

COURSE ABBREVIATION	CSCI 1302
CREDIT HOURS	4 semester hours
COURSE TITLE	Principles of Computer Science II
PREREQUISITES	Completion of CSCI 1301 with a "C" or better

CATALOG DESCRIPTION

The course continues the introduction of the fundamental principles of computer science from CSCI 1301. It extends algorithm development to large programs and introduces additional data structures, pointers, recursion, abstract data types, object-oriented design and programming, algorithm analysis, and file processing techniques, while continuing to emphasize structured programming techniques.

EXPECTED EDUCATIONAL RESULTS

As a result of completing this course, the student will be able to do the following:

1. construct program code to define, set up, read data from, and write data to a data file of type other than text.
2. compare and contrast text files and binary files and describe differences in program code implementation.
3. determine output from a segment of code which uses a recursive subprogram.
4. compare and contrast static and dynamic memory allocation.
5. construct program code to define a pointer variable and use it to create, access, and dispose of a dynamic variable.
6. choose and manipulate appropriate abstract data structures such as lists, stacks, queues, and trees.

7. compare and contrast abstract data types versus objects.
8. demonstrate an understanding of polymorphism and inheritance for objects by using them appropriately in a program.
9. create and use objects and classes appropriately in a program.
10. describe the appropriateness of sorting and searching algorithms in a given context.
11. analyze and determine the efficiency of a specific algorithm using Big-O notation.
12. compare and contrast sequential vs. direct file access and sequential vs. random vs. indexed file organization.
13. describe secondary storage characteristics and how they impact file organization.

GENERAL EDUCATION OUTCOMES

I. This course addresses the general education outcome relating to communications as follows:

A. Students develop their reading comprehension skills by reading the text and handout materials.

B. Students develop their listening skills through lecture and small group problem solving. Lecture material is presented that is not included in the text or handout material and is included as part of the tests or assignments.

C. Students develop their reading and writing skills through the use of problems and activities, including development of computer programs and documentation, developed specifically to enhance their understanding of computer science principles and programming language skills. Students provide written or oral solutions to these problems in either individual or group format. They must also answer short-answer type questions on course exams.

II. This course addresses the general education outcome relating to problem-solving and critical thinking skills through programming assignments that take the student through the programming process from understanding the problem all the way to finalizing a correct program solution to the problem.

III. This course addresses the general education outcomes relating to mathematical concept usage and scientific inquiry as follows:

A. Students apply mathematical concepts in the development of computer programs by creating mathematically-based solutions to the assigned problems and communicating the results of those solutions to the program user.

B. Students apply the scientific method in the set-up and solution of the problems presented to illustrate computer programming principles.

IV. This course addresses the general education outcome relating to organization and analysis of information using a computer by using a modern, fully-capable programming language in the solution of problems designed to illustrate the concepts and principles of computer programming.

COURSE CONTENT

1. Review (5%)
2. Advanced Programming Techniques (20%)
    a) Additional data types (strings, sets, ordinal types)
    b) Recursion
    c) Pointers
    d) Advanced sorting and searching techniques
3. Data Abstraction (35%)
    a) Abstract data types (linked lists, stacks, queues)
    b) Object-oriented design and implementation
4. File processing techniques (30%)
    a) Text vs. Structured component files
    b) Characteristics of secondary storage devices
    c) Access methods (random/sequential/direct)
    d) Fixed-length vs. Varying-length components
    e) Indexed files and key values
5. Analysis of Algorithms (10%)
    a) Characteristics of analysis
    b) Big-O Notation

ENTRY LEVEL COMPETENCIES

Upon entering this course, the student should meet the expected educational

outcomes of CSCI 1301.

ASSESSMENT OF EXPECTED EDUCATIONAL RESULTS

I. COURSE GRADE

The course grade will be determined by the individual instructor using a variety of evaluation methods. The course grade must weigh examinations for at least 50% of the grade and programming assignments for not more than 50% of the grade. Five to seven student programming projects must be assigned. Testing must consist of at least two one-hour examinations and a comprehensive final examination. The final examination must be weighted at not less than 25% nor more than 35%.

II. DEPARTMENTAL ASSESSMENT

CSCI 1301 and CSCI 1302 will be assessed together every five years. The assessment instrument will consist of a common project and a set of free response questions that will be included as a portion of the final examination for all students taking the course.

A committee appointed by the Executive Committee of the Mathematics Academic Group will grade the assessment instrument.

III. USE OF ASSESSMENT FINDINGS

The CSCI Committee, or a special assessment committee appointed by the Executive Committee of the Mathematics Academic Group, will analyze the results of the assessment and determine implications for curriculum changes. The committee will prepare a report for the Academic Group summarizing its finding.

APPROVED DATE: May 2005