

## COURSE OUTLINE

### **Computer Science Information Systems 130 Introduction to Algorithms**

#### **I. Catalog Statement**

CS/IS 130 is a course in programming, algorithm development and problem-solving using both object-oriented and structured approaches. It includes a study of syntax and data structures with applications in science, engineering, and industry. This course is suitable either for students planning to transfer or those wishing to develop a marketable skill.

Total Lecture Units: 3.0

Total Laboratory Units: 0.0

**Total Course Units: 3.0**

Total Lecture Hours: 48.0

Total Laboratory Hours: 0.0

Total Laboratory Hours To Be Arranged: 0.0

**Total Faculty Contact Hours: 48.0**

Prerequisites: CS/IS 112

#### **II. Course Entry Expectations**

Prior to enrolling in the course, the student should be able to:

- examine problems, think in a logical fashion, and provide solutions/algorithms for the problems;
- show the solution/algorithm using flowcharts or pseudocode;
- utilize a compiler to write, debug, and test Java programs.

#### **III. Course Exit Standards**

Upon successful completion of the required coursework, the student will be able to:

- write simple programs using basic unstructured data types;
- use procedures for problem-solving and modularity;
- use objects for data encapsulation;

- develop algorithms and select an appropriate combination of algorithm and data structures for various problems;
- write more complex programs using structured data types, objects, parameters, functions and recursion;
- describe and perform basic operations with reference variables, linked lists, and binary trees;
- convert simple documentation into basic programming including looping and decision;
- refine a program into two or more modules.

**IV. Course Content**

**Total Faculty Contact Hours = 48.0**

- A. Introduction (**4 hours**)
  - 1. Problem definition
  - 2. Object concepts
- B. Basic Concepts (**3 hours**)
  - 1. Data types
  - 2. Scalar data types
    - a. Integer types
    - b. Floating point types
    - c. Character
- C. Elementary Programs (**3 hours**)
  - 1. Elementary programs
  - 2. The assignment statement
  - 3. Program examples
- D. Control Flow (**3 hours**)
  - 1. Looping statements
  - 2. Conditional statements
  - 3. Selection statements
- E. Running, Debugging, and Testing Programs (**6 hours**)
  - 1. Debugging
  - 2. Program testing program
  - 3. Documentation and maintenance
- F. Algorithm Development (**5 hours**)
- G. Additional Data Types (**6 hours**)
  - 1. User - defined
  - 2. Arrays
  - 3. Files
- H. Functions and Methods (**5 hours**)
  - 1. Declaring and invoking a function
  - 2. Local and global variables
  - 3. Parameters – by reference and by value
  - 4. Recursion

- I. **Objects (8 hours)**
  1. Declaring classes
  2. Inheritance
  3. Arrays of objects
  4. Files of objects
- J. **References and Linked Lists (5 hours)**
  1. Reference variables
  2. Stacks and queues
  3. Linked lists
  4. Binary trees

**V. Methods of Instruction**

The following methods of instruction may be used in the course:

- lecture;
- hands-on programming problems using computers.

**VI. Out of Class Assignments**

The following out of class assignments may be used in the course:

- programming assignments (e.g. write a simple program incorporating objects, functions and methods).

**VII. Methods of Evaluation**

The following methods of evaluation may be used in the course:

- quizzes;
- midterm examinations;
- programming projects;
- final examination.

**VIII. Textbook(s)**

Gaddis, Tony. *Starting Out with JAVA: From Control Structures through Objects* 6<sup>th</sup> ed.  
Upper Saddle River: Prentice Hall, 2015. Print.  
11<sup>th</sup> grade Textbook Reading Level. ISBN: 978-0133957051

**IX. Student Learning Outcomes**

Upon successful completion of the required coursework, the student will be able to:

- analyze problems and provide logical solutions;
- outline solutions in a systematic fashion;
- code, test, and debug intermediate level Java programs.