Course Description
EE 2372-Software Design I

2002-04 Catalog: (An introduction to software design with a structured computer language that focuses on the construction of programs consisting of multiple functions residing in multiple files. Covers program creation and top-down-design, basic elements and operations, modular program construction, and the use of programming tools such as makefiles. Introduces object oriented programming techniques. Prerequisite: EE 1305 with a minimum grade of “C” or better.

Prerequisite by Topic: None

Textbook: Three Dialects of C: C, ANSI C, C++ by David H. Williams

Course Outcomes: Students completing EE 2372 will be able to:
1. Design, implement, and execute programs written in the C language.
2. Define and use functions, and design multiple-module programs.
3. Use a variety of unix programming tools for software developments such as debuggers and makefiles.
4. Create and use data structures, including arrays of structures and enumerated data types.
5. Access text files directly in C language programs via I/O functions.

Topics Covered: 1. Introduction, and structure, compilation and execution of C program (4 hrs.)
2. Variables, data types and arrays (3 hrs.)
3. Operators and expressions (4 hrs.)
4. Assignment statements and flow of control statements (4 hrs.)
5. Input and output statements (3 hrs.)
6. Function definitions and function calls (6 hrs.)
7. Structure programming techniques and programming tools (3 hrs.)
8. Pointer definition and use (4 hrs)
9. Derived data structures (4 hrs)
10. File I/O (3 hrs.)
11. Introduction to C++ (4 hrs.)
Class Schedule: Three hours lecture per week

Contribution to Professional Component:
EE 2372 is a required course for all ECE students. This course provides the fundamentals needed for students to take any other software related courses in their curriculum.

Relationship to Program Outcomes:
1. Apply mathematics, science and engineering principles. Students have programming assignments for various applications including sorting algorithms, creation and use of data structures. (Course Outcomes 1, and 4).

3. Have an ability to design a system, component, or process to meet desired needs. Some programming assignments involve design of multiple modules, use of development tools, and accesses to data files. (Course Outcomes 2, 3, and 5).

5. Have an ability to identify, formulate, and solve engineering problems. The assigned programs require students to formulate and then implement algorithms for problem solving. (Course Outcomes 1 and 2).

7. Have an ability to communicate effectively. Each program is required to be well commented. (Course Outcomes 1 and 2).

11. Have an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. Workstations and programming development tools are used in the class. (Course Outcomes 1, 2, and 3).

12. Have a specialization in Computer Engineering. This course focuses on Intel microprocessors, which are widely used in the computer industry. (Course Outcomes 1-5).

Course Outcomes do not related to Program Outcomes 2, 4, 6, 8-10. They do correlate strongly with Educational Objective 1 (30%) and Educational Objective 2 (70%).

Computer Usage: Students use UNIX workstations for lab projects.

Prepared by: Yu-cheng Liu, Professor of ECE