SYLLABUS

DIVISION: Business and Engineering Technology

REVISED: Fall 2014

CURRICULA IN WHICH COURSE IS TAUGHT: IST curricular or elective

COURSE NUMBER AND TITLE: **ITP 100 – Software Design**

CREDIT HOURS: 3 **HOURS/WK LEC:** 3 **HOURS/WK LAB:** 0 **LEC/LAB COMB:** 3

I. CATALOG DESCRIPTION: Introduces principles and practices of software development. Course content includes instruction in critical thinking, problem solving, skills, and essential programming logic in structured and object-oriented design using contemporary tools

II. RELATIONSHIP OF THE COURSE TO CURRICULA OBJECTIVES:

- Implement Information Technology skills required by software applications
- Apply methodologies to stay current in IT offerings, trends, and certifications
- Apply current IT industry standards, protocols, and techniques

III. REQUIRED BACKGROUND: Developmental Requirements if needed ENF and MTE

IV. COURSE CONTENT:

- Program design introduction
- Object-oriented design introduction
- Structured design introduction
- Cooperative team work

V. THE FOLLOWING GENERAL EDUCATION OBJECTIVES WILL BE ADDRESSED IN THIS COURSE. STUDENTS WILL:

X Communication	**X** Critical Thinking
_____ Cultural and Social Understanding	**X** Information Literacy
_____ Personal Development	**X** Quantitative Reasoning
_____ Scientific Reasoning	

VI. LEARNER OUTCOMES	**VII. EVALUATION**
Program design introduction • Understand the place of program analysis and design within the system development life cycle. • Understand the evolution and development of both programming languages and program design. • Describe the difference between a design notational language and a design tool.	Lab exercises Written test
Object-oriented design introduction	Lab exercises

• U Describe a reasonable design process for object-oriented program design. • Explain the use and importance of the Unified Modeling Language within object oriented design. • Identify appropriate tools to use with the standard O-O notational language, the Unified • Modeling Language. • Identify appropriate actors and use cases given a problem description. • Create a UML use case diagram and supporting use case narrative. • Identify appropriate candidate classes given a problem description. • Identify appropriate attributes and operations for candidate classes given a problem description. • Show operations with a full UML signature including received and returned arguments. • Define the terms object, class, message, data-hiding, information-hiding, encapsulation. • Create a UML class diagram. • Explain the use and importance of accessor operations and constructors.	In class assignments Written test
Structured design introduction • Describe a reasonable design process for structured program design. • Identify one or more appropriate notational languages (such as pseudocode) and tools for structured design. • Implement sequence, selection, and loop structures within a structured design solution for an operation. • Implement null ELSE selections, nested selections, and CASE structures within a structured design solution for an operation. • Implement WHILE, UNTIL, FOR loops, and nested loops within a structured design solution for an operation. • Implement structured design solutions that involve one operation calling other operations within the same class and in other classes with received and returned arguments. • Object Model to manipulate the web browser • Understand and use the Document Object Model	Lab exercises In class assignments Written test

Cooperative team work • Work as a member of a design team on a given design task. • Work on a single task of a design project while other teams or individuals are working on separate tasks of the same project.	Program/Presentation