SYLLABUS

DIVISION: Business and Engineering Technology **REVISED:** Fall 2014

CURRICULA IN WHICH COURSE IS TAUGHT: Programming

COURSE NUMBER AND TITLE: **ITP 136 –** C# Programming I

CREDIT HOURS: 4 **HOURS/WK LEC:** 4 **HOURS/WK LAB:** 0 **LEC/LAB COMB:** 4

I. CATALOG DESCRIPTION:

CATALOG DESCRIPTION: Presents instruction in fundamentals of object-oriented programming and design using C# Course content emphasizes program construction, algorithm development, coding, debugging, and documentation of applications within the .NET Framework.

II. RELATIONSHIP OF THE COURSE TO CURRICULA OBJECTIVES

- Acquire the fundamentals of object-oriented programming and design using C#
- Gain an understanding of concepts of object oriented language; encapsulation,
- inheritance and polymorphism
- Learn the C# programming within the .NET framework

III. REQUIRED BACKGROUND: ITP 100 - Software Design

IV. COURSE CONTENT:

- Introduction to Computers and Programming Languages.
- C# language fundamentals.
- Classes and Objects
- Inheritance and polymorphism
- Interfaces
- Arrays and collections
- Strings and regular expressions
- Handling Exceptions

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

_X___Communication
______Cultural and Social Understanding
______Personal Development
______Scientific Reasoning

_X___Critical Thinking
_X___Information Literacy
_X___Quantitative Reasoning

VI. LEARNER OUTCOMES	**VII. EVALUATION**
Introduction to Computers and Programming Languages • Learn the history of computers. • Understand the difference between low level and high level programming languages. • Understand difference between procedural languages and object-oriented language. Understand the basics of computer hardware	Lab exercises In class assignments Project Test
C# Language Fundamentals • Understand types, variables and constants • Understand statements and operators • Be able to work with namespaces	Lab exercises In class assignments Project Test
Classes and objects • Understand how to define a class. • Understand object-oriented concepts of classes • Be able to create objects from the classes • Understand method constructs. • Understand how to pass parameters • Understand the concept of data encapsulation	Lab exercises In class assignments Project Test
Inheritance and Polymorphism • Understand object-oriented inheritance • Be able to describe what polymorphism is and the advantages of it in a language • Understand the concept of abstract classes • Be able to describe boxing and un-boxing	Lab exercises In class assignments Project Test
Interfaces • Be able to describe what an interface is and why it is used • Understand how to implement an interface	Lab exercises In class assignments Project Test
Arrays and collections • Understand array basics • Learn to declare and use one	Lab exercises In class assignments Project Test

dimensional array • Understand the effect of using array elements as parameter to methods • Understand the Collection interfaces	
Strings and regular expressions • Understand C# strings • Understand how to manipulate strings • Be able to describe what regular expressions are and how to use them • Overload output • Overload input	Lab exercises In class assignments Project Test
Handling exceptions • Understand how to throw and catch exceptions • Be able to describe why you would want to use exceptions • Learn how to re-throw exceptions	Lab exercises In class assignments Project Test