Department of Computer Science Course Syllabus

CS 23001  Computer Science II: Data Structures and Abstraction  4 credit hours

Instructor’s Name: ___Prof. Jonathan I. Maletic______________________________

Optional Text Book: Data Structures and Other Objects Using C++, Main and Savitch, Addison Wesley

Course Content:
Computer science concepts and problem solving focusing on data structure and abstraction. Object-oriented concepts and programming including encapsulation, information hiding, object design, generics, polymorphism and an introduction to inheritance. Dynamic memory structures including dynamic arrays, pointers, linked-lists and the use of recursion for problem solving. Abstract data types including stacks, queues, lists, trees and graphs.

Prerequisites or co-requisites: CS 13001 or CS 13012 with a minimum grade of C (2.000). Co-requisite: CS 23022

Required

Goals:
• Continue developing a disciplined approach to problem solving methods and algorithm development.
• Provide a clear understanding of the concepts of abstract data types.
• To teach a number of the basic algorithms and data structures used in computer science.
• To teach the concepts of object oriented programming.
• To provide a foundation for further studies in Computer Science.

Outcomes:
• On completion of this course, students must have a basic understanding of the concepts of abstract data types and object oriented programming methods. Data structures such as lists, stacks, queues, strings, and trees must be understood. The student will have working knowledge of the concepts of classes and objects, operator overloading, constructors, destructors, and generics. The concepts of dynamic data structures and recursion must be well understood.
• PROGRAMING, DESIGN, ANALYSIS

Topics to be Covered:
• Abstract Data Types (ADTs) and the Class Construct
• Classes, Objects, and ADTs
• Software Testing and Set ADT
- Constructors, operator overloading, I/O, Set ADT
- Set ADT + Test Cases
- String ADT, operator overloading
- String ADT, overloading, Assertions, Pre, Post conditions
- Pointers
- Dynamic arrays, copy constructor, destructor
- Copy constructor, destructor, assignment, swap
- Generics, Templates
- Recursion
- Stacks, Stack algorithms, prefix, postfix, infix
- Stack algorithms, Queues
- Linked lists
- Linked lists Stack
- Linked lists Queue
- Double linked lists
- Containers/Iterators
- Containers/Iterators
- Binary trees
- Binary trees
- Backtracking and recursion
- Inheritance, exceptions, virtual