**BSCS - 413: Object Oriented Programming**

**Course Supervisor: Dr. Farhan Ahmed Siddiqui**

### ****Course Description****

This course teaches programming using object oriented programming techniques on windows platform in Java. Students can use their preferred IDE such as Eclipse or NetBeans, but most classroom demonstrations will be in NetBeans. Students will complete several programming assignments to develop their problem-solving skills and to gain experience in detecting and correcting software errors.

### Learning Objective

This course is designed to provide an understanding of object oriented programming concepts: abstraction, inheritance, Encapsulation, polymorphism, overloading etc. It is an informative, entertaining and challenging course. Upon successful completion of this course you will use problem-solving and logic skills to develop Windows applications in Java using object-oriented programming techniques.

**Outline**

**Introduction**  
What is object? What is Object Oriented Programming? UML introduction, Programming Techniques, Characteristics of OOPs, Constructor etc.  
**Introduction to Classes and Objects**   
Class, Objects, Methods and Instance Variables, Declaring a Class with a Method and Instantiating an Object of a Class, declaring a Method with a Parameter, set and get Methods, Access control: public and private.  
**Method overloading & Recursive Method**   
Method overloading, static Methods, static Fields and Class Math, recursion, examples using recursion, Factorial, Fibonacci.  
**A deeper look to Classes and Objects**   
Controlling access to member, Referring to the Current Object’s Members with the this Reference, Overloaded Constructors, default and no argument constructor, Composition, has-a-relationship Enumerations, Garbage Collection, static Class Members, static Import, final instance variable,  
**Data abstraction and Encapsulation**   
Data abstraction and encapsulation, Package access, Creating Packages  
**Inheritance**   
Superclasses and Subclasses, protected Members, Constructors in Subclasses, Object Class.  
**Polymorphism**   
Demonstrating Polymorphic Behavior, Abstract Classes and Methods, Case study: Payroll system using Polymorphism, final Methods and Classes, Downcasting.  
**Interfaces**   
Creating and Using Interfaces, Case study: Payroll System using Interfaces, Declaring Constants with Interfaces, common Interfaces of the Java API.  
**GUI Components**   
Simple GUI-Based Input/Output with JOptionPane, Swing Components, TextFields and Event Handling with Nested Classes, Common GUI Event Types and Listener Interfaces, JButton, JComboBox, JList, JPanel, JTextArea, Key-Event Handling.  
**Generics**   
Generic Methods, Implementation and compile-Time Translation, Overloading Generic Methods Generic Classes.  
**Exception Handling**   
Overview, try catch, finally block, predefine exception handler, Chained Exceptions, declaring new exception types, Pre conditions and Post conditions, java exception hierarchy.  
**Collections**   
Class arrays, Interface Collection and Class Collections, Lists, Array List, Iterator, Linked Lists, Stack, Queue, Sets, Maps,  
**Files and Streams**   
Files and Streams using Class, Class File, sequential access text files, object serialization  
***Design patterns***   
Introduction to design patterns, how design patterns solve design problems? How to select the right pattern and how to use it?

o Object Modeling, Creational Patterns,

o Abstract Factory Builder and Factory method prototype, Singleton

o Structured Patterns: Adapter, Bridge, Composite, Decorator

o Behavioral Patterns: Chain of responsibility, Command, Interpreter, Observer, Iterator, State, Visitor.  
**Accessing Databases with JDBC (Optional)**   
Row Set Interface, Java DB/Apache Derby, Prepared Statements.

**Books**

1. **Java for programmers by Deitel**
2. Introduction to Java Programming, Comprehensive Version by Y Daniel Liang
3. Introduction to Java Problem solving & Programming by Walter Savitch
4. Java Concepts by Cay Horstmann

**Marks Distribution**

Quiz = 10%

Final Examination             = 50%

Assignments                       =10%

Lab / Project = 30%