**OOP LAB**

**Session II**

**Part I & II**

**Lab No. 5:** Strings

Lab Exercises

1. Design a class which represents a student. Every student record is made up of the following fields.
   1. Registration number (int)
   2. Full Name (String)
   3. Date of joining (Gregorian calendar)
   4. Semester (short)
   5. GPA (float)
   6. CGPA (float)

Whenever a student joins he will be given a new registration number. Registration number is calculated as follows. If year of joining is 2012 and he is the 80th student to join then his registration number will be 1280.

Write member functions to do the following.

1. Provide default and parameterized constructors to this class
2. Write display method which displays the record. Test the class by writing suitable main method.
3. Create an array of student record to store minimum of 5 records in it. Input the records and display them.

**Lab No. 6:** Inheritance and Packages

Lab Exercises

1. Create a **Person** class with private instance variables for the person’s name and birth date. Add appropriate accessor methods for these variables. Then create a subclass **College Graduate** with private instance variables for the student’s GPA and year of graduation and appropriate accessors for these variables. Include appropriate constructors for your classes. Then create a class with **main()** method that demonstrates your classes.
2. Define a class Maximum with the following overloaded methods
   1. max (which finds maximum among three integers and returns the maximum integer)
   2. max (which finds maximum among three floating point numbers and returns the maximum among them)
   3. max (which finds the maximum in an array and returns it)
   4. max (which finds the maximum in a matrix and returns the result)

Place this in a package called p1. Let this package be present in a folder called “myPackages”, which is a folder in your present working directory (eg: c\student\3rdsem \mypackages\p1). Write a main method to use the methods of Max class in a package p1.

1. Create an abstract class Figure with abstract method area and two integer dimensions. Create three more classes Rectangle, Triangle and Square which extend Figure and implement the area method. Show how the area can be computed dynamically during run time for Rectangle, Square and Triangle to achieve dynamic polymorphism. (Use the reference of Figure class to call the three different area methods)

**Lab No. 7:**  Interfaces and Multithreading

Lab Exercises

1. Design an interface called Series with the following methods
   1. Get Next (returns the next number in series)
   2. reset(to restart the series)
   3. set Start (to set the value from which the series should start)

Design a class named **By Twos** that will implement the methods of the interface Series such that it generates a series of numbers, each two greater than the previous one. Also design a class which will include the main method for referencing the interface.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\***