CSN-103: Fundamentals of Object Oriented Programming

Assignment 05

General Instructions:

a) Whenever required, use **Scanner** class to accept inputs for the user at the runtime.

b) Follow indentation while writing programs.

c) All submissions will be checked for Plagiarism. ANY ATTEMPT TO CHEAT WILL BE SEVERELY PENALIZED.

Programming Problems

1) Write a Java program which defines an overloaded varargs method **FindElement()** as follows:

   a) If **FindElement()** is called with **variable number** of int arguments then the method should print the value of the smallest arguments.

   b) If **FindElement()** is called with **variable number** of double arguments then the method should print the value of the largest arguments.

   c) If **FindElement()** is called with **variable number** of char arguments then the method should print those arguments in the alphabetical order.

   Example: **FindElement(5,6,2,4,7)** → **Should print 2**
   FindElement(5.0,6.6,2.3,4.9,6.1) → **Should print 6.6**
   FindElement('E','I','Z','M') → **Should print E I M Z**

2) Write a Java program which defines a recursive method **Fibonacci(n)** that returns the n^{th} number in the Fibonacci sequence. The first two numbers in the **Fibonacci** sequence are 0 and 1 (essentially 2 base cases for the recursive method). Each subsequent number is the sum of the previous two numbers, so the whole sequence is: 0, 1, 1, 2, 3, 5, 8, 13, 21 and so on.

3) Write a Java program to emulate the **Queue** data structure [Link]. The queue should be implemented using an int array. Queue can support only two types of operations i.e., **Enqueue()** and **Dequeue()**. The queue should maintain two variables **Front** and **Rear** and these variables must be updated after each Enqueue and Dequeue operation. Print the value of **Front** and **Rear** before and after each Enqueue and Dequeue operation.

   Note:

   a) Declare Enqueue() and Dequeue() as public methods

   a) **Front**, **Rear** variables and int array should be private