UNIVERSITY
OF
JOHANNESBURG

FACULTY OF SCIENCE
ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	CSC03A3/CSC3A10: COMPUTER SCIENCE 3A
CAMPUS	AUCKLAND PARK CAMPUS (APK)
ASSESSMENT	SEMESTER TEST 1 2022

DATE: 2022-03-17	**SESSION:** 14:00 - 16:00
ASSESOR(S):	PROF D.T. VAN DER HAAR MR R. MALULEKA
MODERATOR:	PROF D.A. COULTER DR J. DU TOIT
DURATION: 120 MINUTES	**MARKS:** 100

Please read the following instructions carefully:

1. Answer all the questions.
2. Write clearly and legibly.
3. You may not communicate with other students during the assessment.
4. This paper consists of 3 pages.
5. No calculators are permitted to be used.

QUESTION 1

(a) Analyse the Java source code below and answer the questions that follow: [8]

```
public class Matrix<N> {
  N[][] mat;

   public Matrix(){
     mat = new N[10][10];
   }

   public static void main(String[] args) {
     Matrix<int> test = new Matrix<>();
}
```

1. List two (2) things wrong with the above code [4].
2. How would you fix these problems [4]?

(b) Consider the code below and answer the questions that follow. [10]

```
public static int factorial(int n) throws
    IllegalArgumentException {
   if (n < 0)
     throw new IllegalArgumentException( );
   else if (n == 0)
     return 1;
   else
     return n * factorial(n-1);
}
```

1. Create a iterative function that does the same task called *factorial*.
2. Draw a recursion trace for the recursive factorial(5).

(c) Discuss automatic boxing and unboxing for Wrapper types in Java, and provide a source code example for each. [4]

(d) Give the result/output for the following binary operations: [3]

1. (int)4.3 / (int)2.6
2. 101 | 010
3. 1110 >> 1

Total: 25

QUESTION 2

(a) Provide one reasonable example of an application where using using an array would be better than a singly linked list. [2]

(b) Provide pseudo code that could be used to implement the *removeLast* method of the Doubly Linked List ADT, which removes the last node from the list. [5]

(c) What is the asymptotic relationship between each of the following pairs of functions? [6]

1. $\log n$ and n
2. $3\log_2 n$ and $2\log_3 n$
3. n^k and c^n, where k and c are constants

(d) Which kind of growth best characterizes each of these functions? *Note - You just need to make one tick or cross in the appropriate column.* [6]

	Constant	Logarithmic	Exponential	Polynomial
2^π				
$\log_3 n$				
e^n				
$n^3 + 2n^2$				
π^n				
$n^3 + 2$				

(e) Consider the following function and use primitive counting to determine its runtime as a function of n, the length of the array. Show all assumptions, along with calculations and be sure to provide the final answer in Big-Oh notation. [6]

```

public int calc(int arr[]){
  int n = arr.length;

  for (int i = 0; i < n; i++)
    for (int j = 0; j < n-i; j++)
      if (arr[j] > arr[j+1]){
        int temp = arr[j];
        arr[j] = arr[j+1];
        arr[j+1] = temp;
      }
  }

}
```

Total: 25

QUESTION 3

(a) Consider a Java *Array* of type *T*. Write a class **Stack** that makes use of the Array to realize a **Stack ADT**. *Note - Exception handling does not have to be included.* [10]

(b) Consider the following List Interface and write a class *Queue* that makes use of the List Interface and the Adapter design pattern to realize a *Queue ADT*. **Note: You do not need to implement the List methods.** [10]

```
public interface List<T> {
public Node<T> addAfter(Node<T> elem, T item);
public Node<T> addFirst(T item);
public Node<T> addLast(T item);
public T remove(Node<T> elem);
public Node<T> search(T elem);
public Node<T> first();
public boolean isEmpty();
public Integer size();
}
```

(c) Discuss the Deque ADT, along with one application example where you would use it. [5]

Total: 25

QUESTION 4

(a) Describe the types of **iterators**, along with the limitation of iterators in **Java**. [5]

(b) Provide a complete *Java Class* that realises an *ArrayList ADT*. Your ArrayList should implement the ArrayList Interface listed below and should make use of the **Doubling Strategy**. Your implementation should include all the methods and data members that support the operations below. Your implementation should make use of a **Normal array** (i.e. not a circular array). You may assume that the **shiftElementsRight** and **shiftElementsLeft** methods have been implemented for you, and that all needed packages have been imported. [15]

```
public interface IArrayList<T> {
public T remove(int i);
public void add(int i, T e);
}
```

(c) Discuss the **Sequence ADT** together with an example of where it can be applied. [5]

Total: 25

— End of paper —