CSE 1322 Lecture Test 3O Cover Sheet

Spring 2020

In taking this test you affirm that you have neither given nor received inappropriate help and that the answers you submit are wholly your own.

Print your Name, ID# and NetId on each page.

1) **THERE ARE SIX (6) MAIN QUESTIONS AND AN EXTRA CREDIT QUESTIONS ON THIS TEST. HOWEVER, FOR SIMPLICITY, SOME QUESTIONS ARE SPLITTED INTO MULTIPLE SEPARATE QUESTIONS. PLEASE MAKE SURE YOU ANSWER ALL QUESTIONS!**
2) You have **60** minutes to complete the exam
3) **You cannot use any book, notes, electronic devices, calculator, smart watch, ear phone, etc.**
4) Your code must be exact source code (include all required symbols, syntax, and indentation). It should be written to where a compiler would allow that code to run without any changes from the reader.
5) You are not allowed any electronic devices that can be used to look up or store answers.
6) All answers are to be your own, without the assistance of others
7) Partial credit will be given where appropriate

Student Name : ____

Student KSU ID#_______________________________________

Student NetId :_____________________________________

Student’s KSU ID / NetID: ________________________________ ____________

1) **[10 pts] What is the output from the following code segment?**

```
 public static void Main(){
    int x = 15;
    int y = 25;

    int z = DoIt(ref x, y);

    PRINTLINE(x + “ “ + y + “ “ + z);  // PRINTLINE 1
 }

static int DoIt(ref int a, int b)
    {
      a += 25;
      b -= 15;
      PRINTLINE(a + “ “ + b); // PRINTLINE 2
      return (a + b);
    }
```

Output from PRINTLINE 1:

Output from PRINTLINE 2:

Student's KSU ID / NetID: ______________________________ __________

2) [20 pts] Suppose you have the following **singly linked list** Node and LinkedList classes, answer the following three questions. **You don't need to write the classes again, just complete the requirements.**

```
public class Node {
      int id;
      Node next;
      public Node(int id){ this.id = id;       }
}
public class LinkedList{
      Node head;
      LinkedList(){   head = null;  }
      public boolean isEmpty(){ return(head == null);   }

public void append(int id)
{ /* Codes go here */ }

public void prepend(int id)
{ /* Codes go here */ }

} // End of the LinkedList class
```

(i) Suppose, you have the following codes in your main() methods. The write the output sequence of the node ids, separated by a comma, if displayed on a console, as correct order.

Hints: The output should be something as follows: 101, 102, 103, 104, 105 **All in one line.**

```
LinkedList ll = new LinkedList();

ll.prepend(101);
ll.append(102);
ll.prepend(103);
ll.append(104);
ll.prepend(105);
```

Student's KSU ID / NetID: ________________________________ ___________

(ii) **Considering an initial empty LinkedList object,** write only the common codes in the **`prepend(int id)`** and **`append(int id)`** methods using Java or C# (**no pseudocode allowed**);

Student's KSU ID / NetID: ______________________________ __________

3) [15 pts] Suppose, you have the following Node class, using C# or Java (**no pseudocode allowed**), write the code for the **pop()** method for a **Stack**. **You don't need to write the classes again, just complete the requirements.**

```
public class Node {
    int id;
    Node next;
    public Node(int id){ this.id = id; }
    }  // End of the Node class

public class Stack {
    Node top;
    Stack(){  top = null;  }
    public boolean isEmpty() { return (top == null);  }

    public void push(int id)
    { /* Codes go here */ }

    public void pop()
    { /* Codes go here */ }

    } // End of the Stack class
```

pop() method:

Student's KSU ID / NetID: ________________________________ ___________

4) [15 pts] Suppose, you have the following Node class, using C# or Java (**no pseudocode allowed**), write the code for the **enQueue(String name)** method for a **Queue**.

```
public class Node {
     String name;
     Node next;
     public Node(String name){ this.name = name;      }
}

public class Queue{
     Node head;
     Node tail;
     Queue(){   head = null;  tail = null;  }
     public boolean isEmpty(){ return( head == null);  }

public void enQueue (String name)
{ /* Codes go here */ }

public void deQueue()
{ /* Codes go here */ }

} // End of the Queue class
```

enQueue(String name) method:

Student's KSU ID / NetID: ______________________________ ___________

5) [20 pts] Using your choice of language, C# or Java (**no pseudocode allowed**), write a recursive method named **recursiveDisplay(int n)** that displays the squared of integer numbers from 1 to n in Ascending order, separated by at least a single space between the squared-numbers. You may **not** use global variables or stacks.

Hints: The output should be something as follows: 1 4 9 16 25 . . . n*n

recursiveDisplay(int n) method:

Student's KSU ID / NetID: ______________________________ ___________

6) [20 pts] Using your choice of language, C# or Java (**no pseudocode allowed**), give a format / template of Exception Handling codes **with try, catch, and finally blocks**. In the try block include some statements that raise **an arithmetic exception** and the catch block catches that arithmetic exception. You don't need to write any complete method or program, just write the code segments with try, catch, and finally blocks.

7) [2 points] Extra Credit Question [Circle only one]:

Is factorial of 0 equal to factorial of 1? Yes / No