THE UNIVERSITY OF MANITOBA
COMP 1020: Introductory Computer Science 2
Term Test 2 (TT2)
A01 – Thursday, November 20, 2025

Student Information: Fill in one letter or number per box.

Academic Integrity Contract

I understand that cheating is a serious offence.

> *"As members of the University Community, Students have an obligation to act with academic integrity. Any Student who engages in Academic Misconduct in relation to a University Matter will be subject to discipline*." (2.4 - Student Academic Misconduct Procedure).

Student Signature:_____

Instructions – Please read carefully

1. Answer all questions on this paper ***in the spaces provided***. Answers must be on the same page as the question. Material on the cover page, a blank page, or on the wrong page, ***will not be marked***.
2. Answer questions using **Java programming language** and use **appropriate technical terms**.
3. Tests will be scanned for grading. **Write darkly**, and do not write very close to the edge of the paper, or in the **top** margin.
4. No aids ***of any kind*** (such as calculators, language translators, phones, smart watches, etc.) are permitted.
5. You have 60 minutes to complete the test.
6. There are 9 questions, and a total of 30 marks.
7. This test is worth 11% of your final grade in COMP 1020.
8. Fill in your identification information above. ***Do not*** write this information on any other page.

Part 1: Short Answer Questions (14 marks)

1. [3 pts total] For each of the following statements, indicate if the statement is True or False AND do the following.

- Correctly identify: **a True statement (1 mark), a False statement (0.5 marks).**
- **If the statement is False, indicate <u>why</u> (0.5 marks)**. You do not need to rewrite the entire statement; you can cross out words and replace them with the correct words or add words to the statement. <u>You do not earn any marks for simply negating the statement</u> (e.g., changing "This is a question." to "This is not a question.").

a) [1 pt] **`myArrayList.add(1,"words");`** updates the element at index 1 in an ArrayList.

b) [1 pt] `MergeSort alternates between left and right sides of an array before merging sides back together.`

c) [1 pt] Insertion sort performs worst on an already sorted list.

2. [1 pt] Other than the lack of comments, identify at least two programming standard violations in the code below.

```
class Example {
     public static void Main(String[] args) {
     System.out.println( "Hello");
         if (true) {
             System.out.println( "World"); }
     }
 }
```

3. [2pt] What is the output of the code below?

```
public class Main {
    public static int process(int i, int j) {
        if (j <= 1) return 5;
            return i + process(i * 2, j - 1);
    }
    public static void main(String[] args) {
        System.out.println(process(10, 4));
    }
}
```

4. [2 pts] Describe what the following code is doing and name the data structure it belongs to.

```
public void example() {
    if(head != null && head.next != null) {
        if(head.next.next != null) {
            head.next.next = head.next.next.next;
        }
    }
}
```

5. [2 pts total] Consider the following MD integer array declaration:

```
int[][][][] numbers = new int[2][2][2][2];
```

a) [0.5 pts] How many total int values can be stored in this array? _____

b) [1.5 pts] Draw the memory diagram/representation of the multidimensional array. Include stack, heap, array(s) as objects, reference arrows, and any default values.

6. [3 pts total] Answer the following two questions given to code below:

```
public class Main {
    public static void sort(int[] arr) {
        for (int i = 1; i < arr.length; i++) {
            int j = i - 1; int target = arr[i];
            while (j >= 0 && arr[j] < target) {
                arr[j + 1] = arr[j]; j--;
            }   arr[j + 1] = target;
        }
    }
}
```

a) [1 pt] Name the sorting algorithm used in the code.

b) [2 pts] Demonstrate how the algorithm works by providing a step-by-step walkthrough of sorting the array: [8, 7, 3, 6, 1, 9].

Ensure that your diagram captures the state of the array at the end of each iteration of the outer for loop (i.e., each 'pass' of the algorithm).

7. [1 pt] Which of the following is true about merge sort? Circle or indicate all that apply:

A. Two sorted arrays are provided to the merge step.

B. Merge sort is less efficient than other sorting algorithms we have learned so far.

C. Merge sort can only be implemented recursively.

D. Merge sort uses the divide and conquer technique.

Part 2: Long Answer Questions (16 marks)

8. [8 pts total] The double factorial of a non-negative integer, n, is defined as follows:

- If n is even: **`n!! = n x (n-2) x (n-4) x … x 2`**
- If n is even: **`n!! = n x (n-2) x (n-4) x … x 1`**
- By convention, **`0!! = 1`**.

a) [5.5] Write a recursive method that takes an integer n and returns the double factorial of n, according to the definition above. Follow best programming standards including method declaration, variable naming, and handling invalid input.

b) Example:

Input: 5 Result: 15

Input: 8 Result: 384

Input: 1 Result: 1

b) [2.5 pts] Provide the input (parameters) to your method which will result in the method being called 3 times (total), including the first time with the original input (i.e., depth = 3). Show what the input and output is for each call. Call #3 should result in the base case being triggered.

Call #1 input: _____

Call #2 input: _____

Call #3 input: _____

Base Case Reached!

Output #3: _____

Output #2: _____

Output #1: _____ (final result)

9. [8 pts] Write a method that will be passed in a 2D jagged array of integers and modify it so the value at each location is the sum of the values in the previous row adjacent to it (i.e., the values above and to upper diagonal corners). The first row and first column of the array contain all 1s and your method will fill the rest.

Example:

Input:

1 1 1 1 1 1

1 3 3 3

1 7 9 6 3 0

1 17 22

Blank Page for continued work or scrap notes.

If you are continuing work, clearly indicate which question is continued on this page and where.

Use this page for rough work only. It will NOT be graded.