

**Department of Computer Science
Graduate Exam in Programming
Spring 2020**

You must write **Java** or **C++** on the first page of your answers!! You may answer the questions using either Java or C++, but you must use the same language for **each** question on the exam.

1. (2 pts) Write a function named `triangle` to output the following pattern:

```
 *
  **
 ***
****
*****
*****
*****
```

Your function should have one parameter, `x`, which denotes the number of rows (and columns) to be printed. The pattern above is for `x=7`.

2. (1.5 pts) Write a **recursive** function named `sumOver` that takes an `int` parameter `n` and returns a `double` value which is the sum of the reciprocals of the first `n` positive integers. The reciprocal of `x` is $1.0/x$. For example, `sumOver(3)` returns `1.833` which is $1.0/1 + 1.0/2 + 1.0/3$. Your function must work for values of `n` greater than or equal to 0. By definition, `sumOver(0) = 0`.

Do not use loops, extra parameters, or global or static variables

3. (2.5 pts) Implement the following two functions that operate over arrays of `int` (assume the values in the array are unique):

A. `indexOfMax(array, size)`: This function should return the index of the largest element of the array (assume that the size is greater than 0).

B. `selectionSort(array, size)`: This function should sort the elements of the array using the following algorithm: find the largest element and move it into the last position, then find the next largest element and move it into the second to last position, and so on, until the entire array is sorted.

—Do not use any other algorithm to sort the list.

—This function should call the `indexOfMax` function from part A.

4. (4 pts) Declare and implement a class to represent a **stack** of integers, called `IntStack`. Include the following functions in your class:

- a default constructor that creates an empty stack.
- a void function for `push(x)` that inserts a new value onto the top of the stack.
- an int function for `pop()` that removes the value from the top of the stack, and returns it (if the stack is empty, it returns -1).
- a boolean function `isEmpty()` that returns true if the stack is empty, otherwise false.

Note that a stack adds and removes elements in “Last In, First Out” order. The most recently added element is the one that is removed next.

Implement the class functions using a **linked list** composed of Nodes to store the values in the stack. Hint: `push(x)` should add `x` to the front of the list and `pop` should remove the element at the front of the list. Use the following declarations in your class:

| | |
|--|---|
| <pre>// C++ private: struct Node { int value; Node *next; };</pre> | <pre>// Java private class Node { int value; Node next; }</pre> |
|--|---|