CS 2150 Exam 2, spring 2022

Name _____

You MUST *clearly* write your name above. You must also write your e-mail ID on **EACH** page.

If you are still writing when "pens down" is called, your exam will not be graded – even if you are still writing to fill your name and userid. So please do that first. Sorry to have to be strict on this!

There are 6 pages to this exam. Once the exam starts, please make sure you have all the pages. Questions are worth different amounts of points.

Answers for the short-answer questions should not exceed about 20 words; if your answer is too long (say, more than about 30 words), you will get a zero for that question! This is not a hard maximum, but you should try your best to answer the question as concisely as you can.

This exam is CLOSED text book, closed-notes, closed-calculator, closed-cell phone, closed-computer, closed-neighbor, etc. Questions are worth different amounts, so be sure to look over all the questions and plan your time accordingly. Please sign the honor pledge below.

_____

_____

_____

_____

You step in the stream,
But the water has moved on.
This page is not here.

Page 2: Exam 1 material

1. [3 points] What is the big-Theta, worst-case runtime of the *Vector resize() method*. *Briefly* explain your answer.

2. [3 points] *Briefly,* what does the following C++ code do? Clearly state whether it *always seg-faults, always runs to completion, or you don't know which*. Also, explain your reasoning.

```
int main(){
    int* ptr;
    int z = 56;
    cout << *ptr << endl;
    ptr = new int(7);
    cout << *ptr << endl;
    ptr = &z;
    cout << *ptr << endl;
    return 0;
}
```

3. [6 points] Rank the following in order of largest to smallest range of possible values. The *range* of a numerical type is the maximum value minus the minimum value. For example, a 4-bit unsigned integer (aka a nibble) has a range of $15-0=15$. If there is a tie, the order you rank them does not matter. *Your answer can be the letters presented below, from largest range to smallest range (e.g., A,B,C,D,E,F).*

   A) 4 byte (32 bit) unsigned integer
   B) IEEE-754 floating point
   C) 8-byte signed long
   D) 32-bit floating point with 1 sign bit, 12 exponent bits, and 19 mantissa bits
   E) IEEE-754 double (1 sign bit, 11 exponent bits, 52 mantissa bits)
   F) 32-bit two's complement integer

Page 3: Trees

4. [3 points] Given a *Binary Search Tree*, write a method that will print out the elements in descending order. *Briefly* describe (at a high-level) how you would approach this. We are looking for an English explanation here, not pseudo-code.

5. [3 points] You have an AVL Tree with an imbalanced node x. List the balance factors x and its two children given that a *double left-right rotation* (left rotation first, then right) is needed to fix the tree. If you cannot know the balance factor for a node, simply write *do not know*.

6. [6 points] You are given a pointer to the root of a *BST* in which one node violates the BST ordering property. Fill in the blanks in the method below. *Hint: Maintain a range (lower and upper) that all values in a subtree should be within and update the range as you traverse the tree*

```
/* Initially called with root, lower=-INF, and upper=INF */
/* Returns pointer to bad node, or NULL if not in this subtree */
public TreeNode* findBadNode(TreeNode *curNode, int lower, int upper){

    if (curNode == nullptr)  ____________________; // base case (not found)

    if (_________________ || _________________) return curNode; / /found it

    TreeNode* left = findBadNode(curNode->left, __________, __________);

    TreeNode* right = findBadNode(curNode->right, __________, __________);

    if (left != nullptr) return left;
    return right;
}
```

Page 4: Hashes

7. [3 points] Suppose we are unconcerned with memory usage, but want our *hash table* to run very fast. *Briefly* list two ways you could alter a hashtable to increase the average access speed.

8. [3 points] Suppose you are using *double hashing* as your collision resolution strategy and have decided to use the following secondary hash function: $h_2(x) = (3x+1)\%9$. *Briefly*,what is the problem with using this function (we are looking for a *major flaw*)?

9. [6 points] Consider a hash table of size 10:

index	value
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

The keys are non-negative integers, and the hash function is $hash(k) = k\%10$. Insert the elements listed below into the table, and use *linear probing* as your collision resolution strategy. The keys to insert are: 45, 27, 14, 24, 86.

Page 5: Assembly & IBCM

10. [3 points] In your own words, *briefly* describe why we used the *doit* approach to iterating through arrays in *IBCM*. By *doit*, we mean the process of computing some arithmetic result, storing that directly in our program, and then executing that line later. What was the purpose and why did we need to do this?

11. [3 points] Rewrite the x86 command `pop rax` in two equivalent lines that don't use the `pop` opcode

12. [6 points] The following x86 method accepts an array, and the size of that array. The method should return the largest value in the array. Fill in the missing blanks to make this method operate correctly.

```
mystery:    /* Zero-out eax, r10d, r11d */
            mov     eax, [edi]
body:       cmp     r10d, esi

            je      _______
            mov     r11d, [edi+4*r10d]

            inc     _______
            cmp     eax, r11d

            jl      ________
            jmp     body
update:
            mov     ______, r11d
            jmp     body
done:       ret
```

Page 6: Nothing to see here

This page unintentionally left blank.

Move along, move along.