The purpose of this first problem set is to remind you of the material you need from 2574 and help get you up-to-speed with python. This course uses these concepts from the first few lectures through the end of the semester, so it is important that you understand this material.

You can complete the exercises by either directly marking up this pdf, or by printing, completing, and scanning as a pdf. The resulting pdf file should be uploaded to scholar via the assignment tab by the due date and time.

---

**Exercises**

1. Consider an abstract data type (ADT) representing a stack (LIFO) of integers. Determine the stack contents at the points indicated below during the following operations. Write down the stack contents after the operation on the given line is executed. Be sure to indicate the top of the stack.

```python
1 stack<int> s;
2 s.push(2)
3 s.pop()
4 s.push(5)
5 s.push(3)
6 s.pop()
7 s.pop()
8 s.push(4)
9 s.push(3)
10 s.pop()
11 s.push(1)
```

(a) After line 1: $\varepsilon \varepsilon \varepsilon$
(b) After line 3: $\varepsilon \varepsilon \varepsilon 2$
(c) After line 5: $\varepsilon 2 3 5$
(d) After line 8: $\varepsilon 3 2 5$
(e) After line 11: $\varepsilon 3 2 5$

2. Consider an ADT representing a queue (FIFO) of integers. Determine the queue contents at the points indicated below during the following operations. Write down the queue contents after the operation on the given line is executed. Be sure to indicate
the front and back of the queue. The operation add(x) adds the argument x to the
queue, while the operation remove removes the next item from the queue.

1 queue<int> q;
2 q.add(2)
3 q.remove()
4 q.add(5)
5 q.add(3)
6 q.remove()
7 q.remove()
8 q.add(4)
9 q.add(3)
10 q.remove()
11 q.add(1)

(a) After line 1: ῥξ
(b) After line 3: ῥξ
(c) After line 5: 3, 5 ῥξ
(d) After line 8: 4 ῥξ
(e) After line 11: 1, 2, ῥξ

3. Consider an ADT representing an priority queue of strings. Determine the queue
contents at the points indicated below during the following operations. Write down
the queue contents after the operation on the given line is executed. Be sure to indicate
the front of the queue. The operation add(x, p) adds the argument x to the queue with
an integer priority of p, while the operation remove removes the next item from the
queue. Integers with lower numeric value are interpreted as having a higher priority.
Items with identical priorities are treated in FIFO order.

1 priority_queue<string, int> pq;
2 pq.add("walk", 10)
3 pq.remove()
4 pq.add("run", 1)
5 pq.add("run", 2)
6 pq.remove()
7 pq.remove()
8 pq.add("walk", 10)
9. \texttt{pq.add("walk", 5)}
10. \texttt{pq.remove()}
11. \texttt{pq.add("walk", 1)}

(a) After line 1: `empty`
(b) After line 3: `empty
(c) After line 5: `\text{\small \{walk, 1\}} (\text{\small \{walk, 2\}}`
(d) After line 8: `\text{\small \{walk\}} (\text{\small \{walk, 10\}}`
(e) After line 11: `\text{\small \{walk\}} (\text{\small \{walk, 10\}}`

4. Consider an ADT representing a binary sort tree (BST). Determine the Binary Search Tree contents after the following operations. Consider the keys as the stored item. Be sure to indicate the root of the tree.

\begin{verbatim}
BST t;
t.insert(51);
t.insert(10);
t.insert(5);
t.insert(75);
t.insert(64);
t.insert(57);
t.delete(64);
t.delete(5);
t.delete(51);
t.insert(69);
t.insert(80);
t.insert(100);
t.insert(1);
t.insert(-4);
t.insert(15);
t.insert(25);
\end{verbatim}

5. For your resulting tree in Exercise 4, determine the order (printing keys left-right) of the following traversals.

(a) Preorder: `57, 10, 1, -4, 15, 25, 75, 64, 80, 100`
(b) Inorder: `\text{\small -4, 1, 10, 15, 25, 57, 64, 75, 80, 100}`
(c) Postorder: `\text{\small -4, 1, 15, 25, 10, 64, 100, 80, 75, 57}`