UNIVERSITY OF THE FRASER VALLEY

COMP 251 – MIDTERM EXAMINATION
May 24th, 2017

KEY

INSTRUCTIONS

1. Calculators are **not** permitted.
2. This exam is **closed book**.
3. Clearly print your name and student ID number on this examination (above).
4. This exam contains **multiple choice** questions.

   NOTES REGARDING MULTIPLE CHOICE QUESTIONS:
   - There are **five** possible choices per question.
   - There is one **best** choice for full credit (+1).
   - The remaining four choices are **worthless**.

5. The values of all non-multiple choice questions are stated explicitly in **bold**.
6. There are **40 points** in total.
7. There are **6 pages** including this cover sheet.
8. You have **90 minutes**.
9. Dictionaries are NOT allowed.

1. **[4]** Describe in detail the meaning of big-*O* for an arbitrary function, *f*(*n*), and then state its formal definition. Describe the meanings of big-Ω, and big-Θ notation:

> **Memorize:** Suppose $f : \mathbf{Z} \to \mathbf{R}$ and $g : \mathbf{Z} \to \mathbf{R}$ are functions. We say f is $O(g)$ if there exists constants C and k so that $|f(n)| \leq C|g(n)|$ for *all* $n > k$.
>
> In other words, f is $O(g)$ if it is never larger than a constant times g for all large values of n. The function $Cg(n)$ gives an upper bound on the size of $f(n)$ for all large values of n. Usually the expression for g is less complex for the expression for f, and that's one of the things that makes big-O notation useful. Notice that we don't care what happens for "small" values of n. Also, usually we don't worry too much about the absolute value signs since we usually compare functions that take positive values.

2. **[2]** Suppose an algorithm for processing a retail store's inventory takes 10^4 ms to read the initial inventory from disk and then 10 ms to process each transaction (items acquired or sold). What is the *upper bound on the time complexity* of this algorithm?

   (10^4 + 10n) ms so the algorithm is big-*O* (n)

3. **[5]** Prove that $f(n) = 5n^2 - 2n + 16$ **not** $O(n)$.

> **Example 2.** *Prove that* $5n^2 - 2n + 16$ *is not* $O(n)$. Assume $5n^2 - 2n + 16$ is $O(n)$. Then there exist constants C and k so that $5n^2 - 2n + 16 \leq Cn$ for all $n > k$. Dividing both sides by n (and assuming $n > 0$) we get $5n - 2 + 16/n \leq C$, or $n \leq C + 2 - 16/n \leq C + 2$. This inequality does not hold for $n > C + 2$, contrary to our assumption that it held for all large values of n. Therefore $5n^2 - 2n + 16$ is not $O(n)$.

4. **[3]** Provide a big-*O* analysis of the following code fragment:

```
int sum = 0;
for (int i = 0; i < n; i++)
   for (int j = 0; j < n; j++)
       sum++;
```

```
1          int sum = 0;
1+2*n      for (int i = 0; i < n; i++)
n*(1+2*n)  for (int j = 0; j < n; j++)
n*(1*n)    sum++;
```

3*n^2 +3*n +2 instructions

O(n^2)

5. Which of the following statements is a valid array declaration?

```
6.    int[] a = new int(30);
7.    char[] b = new char[0];
8.    int[] c = new int[][3];
9.    int[][] d = new int[5][];
```

6. What is the value of a[1]?

```
7.    int[] a = new int[5];
```

7. What is the value of b[0]?

```
8.    int [] a = {1,2,3};
9.    int [] b = (int[])a.clone();
10.   if (a == b) b[0]++;
```

8. What is the value of b[0]?

```
9.    int [] a = {1,2,3};
10.   int [] b = (int[])a.clone();
11.   if (Arrays.equals(a,b)) b[0]++;
```

9. What is the value of twin[1]?

```
10.   Object[] obj = {new Integer(10), new String("CMU"), new
   Double(1.23)};
11.   Object[] twin = (Object[]) obj.clone();
   obj[1] = new Integer(15);
```

ANSWERS

1. Which of the following statements is a valid array declaration?

```
int[] a = new int(30);
char[] b = new char[0];
int[] c = new int[][3];
int[][] d = new int[5][];
```

2. What is the value of a[1]?

```
int[] a = new int[5];
```

0

3. What is the value of `b[0]`?

```
int [] a = {1,2,3};
int [] b = (int[])a.clone();
if (a == b) b[0]++;
```

1

4. What is the value of b[0]?

```
int [] a = {1,2,3};
int [] b = (int[])a.clone();
if (Arrays.equals(a,b)) b[0]++;
```

2

5. What is the value of `twin[1]`?

```
Object[] obj = {new Integer(10), new String("CMU"), new Double(1.23)};
Object[] twin = (Object[]) obj.clone();
obj[1] = new Integer(15);
```

"CMU"

10. **[3]** What is a Set? List? Map?

A set is a data structure that contains at most one of any element.

A list is a data structure that contains any number of any element.

A map is a data structure that contains (key, value) pairs where a key is assigned to at most one value.

11. **[2]** What is an iterator and why are iterators necessary?

An iterator is a type of object that is used to traverse a data structure.

12. **[3]** Given an array and a singly linked list. Which of these data structures uses more memory space to store the same number of elements? Explain your answer.

A singly linked list, because it also needs to maintain the pointers to the other nodes, which takes up space.

13. What changes do you need to make to a linked list in order to have a *constant time access* to the last node?

    The linked list should have a tail pointer in addition to a head pointer.

14. What is the worst-case complexity of searching in a linked list with n nodes?

    a) O(1)
    b) O(log n)
    c) O(n)
    d) O (n^2)

15. What is the worst-case complexity of merging two linked lists with n nodes?

    a) O(1)
    b) O(log n)
    c) O(n)
    d) O (n^2)

16. How many references must you change to delete a node from the middle of a singly linked list?

    a) 1
    b) 2
    c) 3
    d) 0

17. Why might one choose to use a singly linked list instead of a doubly linked list?

    a) Insert is not efficient for a doubly linked list
    b) A doubly linked list has a fixed size
    c) Memory usage is a big concern for you
    d) Remove takes constant time for a singly linked list

18. A Queue is best characterized as
    a) Last In First Out
    b) First In Last Out
    c) First In First Out
    d) None of the above

19. Given an empty queue Q, what does it look like after the following operations?

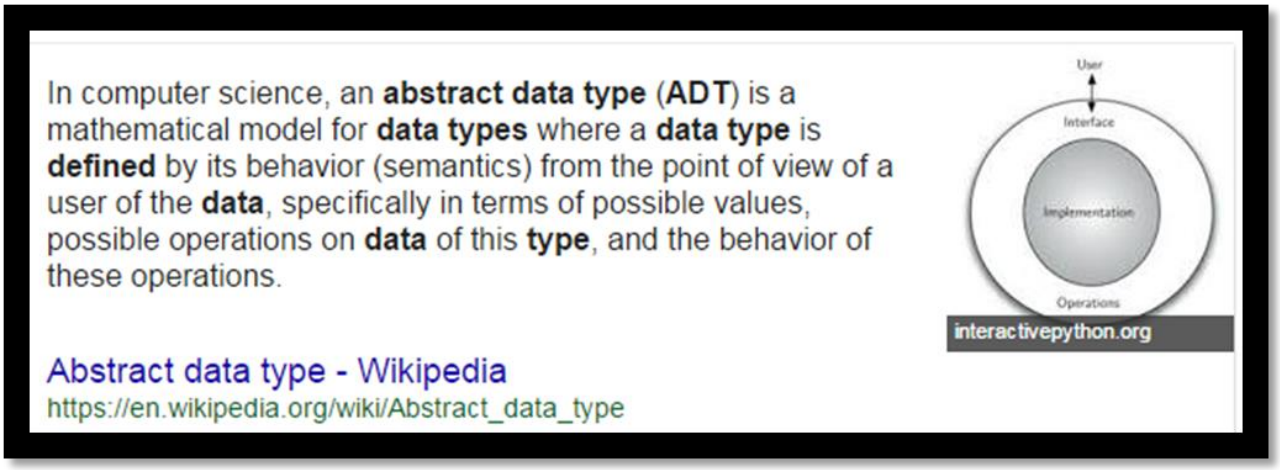
```
Q.enqueue(5)
Q.enqueue(2)
Q.dequeue()
Q.enqueue(3)
Q.dequeue()
```

a) 3
b) 5
c) 9
d) none of the above

20. Given a 5 element stack S (from top to bottom: 2, 4, 6, 8, 10), and an empty queue Q, remove the elements one-by-one from S and insert them into Q, then remove them one-by-one from Q and re-insert them into S. S now looks like (from top to bottom).
a) 2, 4, 6, 8, 10
b) 10, 2, 4, 6, 8
c) 10, 8, 6, 4, 2
d) none of the above

21. What is the reason for using a "circular queue" instead of a regular one?
a) the running time of enqueue() is improved
b) reuse empty spaces
c) you can traverse all the elements more efficiently
d) None of the above

22. **[2]** Define an ADT:



23. **[3]** Perform Breadth-First Traversal of the following directory tree using a queue: