Name:

CMPT 202 Quiz 2
October 13, 2019
55 Total Points

Multiple choice: Choose the best answer(s) and write an "X" next to them.

1. (3 pts) The specifications of an ADT's operations in an interface indicate _____.
   a. What the operations do, along with the input and output of each operation.
   b. How to implement the operations
   c. How to store the data
   d. All of the above

2. (3 pts) Which of the following will be true when the reference variable `current` references the last node in a linked list with > 1 element?
   a. `current == null`
   b. `head == null`
   c. `current.next == null`
   d. `head.next == null`
   e. none of the above

3. (3 pts) In a linked-list implementation of the List interface _____.
   a. the `add()` operation will sometimes require copying the entire list
   b. one node explicitly references the next node
   c. removing an item requires shifting all remaining elements in the list
   d. none of the above

(3 pts) What is the Big-oh notation of the following algorithm?

```
double oneOver = 0.0;

for (int i = 1; i < (n - 1); i++) {
 oneOver += (1.0/i);
}
```

a. O(n)

b. O(1)

c. $O(n^2)$

d. $O(2^n)$

5. (3 pts) Values in a stack are returned in _____ order.

a. First-in First-out (FIFO)

b. according to a specified position

c. Last-in First-out (LIFO)

d. Priority

6. (3 pts) Assuming a linked list is being used and that `previous.next` is `current`, which of the following statements removes the node represented by `current`?

a. `previous.next = current;`

b. `current.next = previous;`

c. `current.next = current.next;`

d. `previous.next = current.next;`

7. (3 pts) Which of the following could be used to implement a Stack?

a. Set

b. Linked list

c. Java ArrayList

d. Array

8. (3 pts) Which of the following could be used to implement a Queue?

a. Set

b. Linked list

c. Java ArrayList

d. Array

9. (2 pts) Write the contents of the stack after the following operations. Be sure to indicate what is on top of the stack.

Push(A)
Push(B)
Push(C)
Peek()
Pop()
Push(D)

Top ----------> Bottom

Your answer here:

10. (3 pts) Values in a queue are returned in _____ order.

a. First-in First-out (FIFO)

b. Queues have random access

c. Last-in First-out (LIFO)

d. Priority

11. (2 pts) Show the contents of the queue after the following operations. Be sure to indicate what is the front and rear of the queue.

Add(A)
Add(B)
Add(C)
Remove()
Remove()
Add(D)
Remove()
Add(E)

Front ----------> Rear

Your answer here:

12. (10 points) Write a method called `printList()` that outputs the values in a linked list. Each node in the list is represented as follows:

```
public class Node<T>
{
     T data;

     Node next;

}
```

Assume `head` refers to the first node in the list and is used to call your method as follows:

```
Node temp = head;
printList(temp);
```

Your method begins below

```
public void printList(Node<T> start) {
```

13. (14 points) Assume you have the following `Fraction` interface:

```
public interface Fraction {
  /**
   * Sets the value of the Fraction to the value n/d.
   */
  public void set (int n, int d);

  /**
   * Accessor for the numerator.
   */
  public int getNumerator();

  /**
   * Accessor for the denominator.
   */
  public int getDenominator();

  /**
   * Returns the value of this Fraction as a double
   */
  public double getValue();

  /**
   * Adds a Fraction other to this Fraction
   * Returens the sum (this + other)
   */
  public Fraction add (Fraction other);

}
```

Now assume you have two classes, `Rational` and `MixedNumber`, that both implement the `Fraction` interface. (The actual details of how the interface is implemented is unimportant.)

```
public class Rational implements Fraction {
    public Rational(int num, int denom)
   ...
}

public class MixedNumber implements Fraction {
    public MixedNumber(int whole, int num, int denom)
   ...
}
```

(Assume all the following code appears in a main method in a separate class.)

Indicate all the lines that are correct (*i.e.*, compile and execute *without* errors): (4 points)

```
Fraction oneThird = new Rational(1,3);

MixedNumber oneThird = new Fraction(1,3);

Fraction oneHalf = new MixedNumber(0,1,2);

MixedNumber oneHalf = new Fraction(0,1,2);
```

Assuming you are able to successfully create `oneThird` and `oneHalf` objects, write code to:

- (4 points) add `oneThird` and `oneHalf`
- (4 points) save the result in a variable called `sum` (don't forget to declare `sum`)
- (3 points) print the value of the variable (using `getValue`)

You can assume this code should be part of a `main()` method. You **do NOT** need to implement the `add()` method, just call the method.