cs3250s22
quiz4 for jmn8gf
Jason Nguyen

Your original score on this exam was 7 of 10 points (70.00%).

Your breakdown of points per page is below.

Page	Score	Max
1	0	0
2	0	0
3	7	10
4	0	0

A graded scan of each page follows.

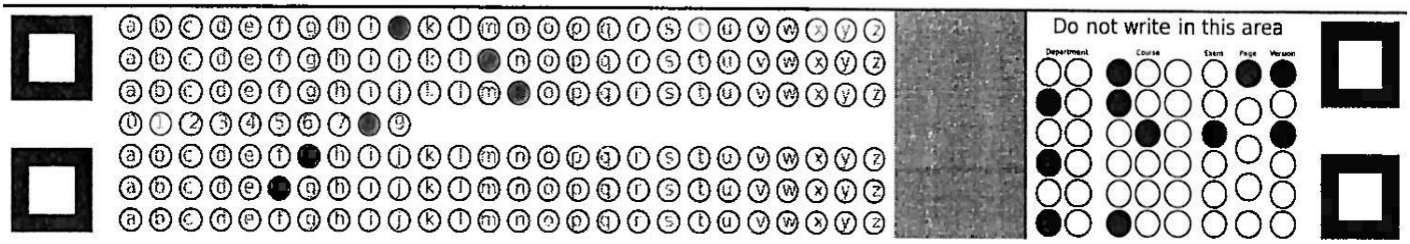
CS 3250: Quiz 4

28-March-2022

Name of your **left** neighbor (mark X if there is none)	**Your name**	Name of your **right** neighbor (mark X if there is none)
Annie Hua	Jason Nguyen	Quang Lam
	Your computingID	
	jmn8gf	

Bubble your ComputingID.

(-1 pt. if your computingID is not bubbled or not bubbled properly. Be sure to bubble the entire circle)



Instruction: Answer the questions as concisely as you can. Please **write neatly** and **in the space provided**; if we can't read it, we have to mark it wrong.

Consider the `min()` method

```
public static <T extends Comparable<? super T>> T min (List<? extends T> list)
{
   Iterator<? extends T> itr = list.iterator();
   if (itr.hasNext() == false)
      throw new IllegalArgumentException("min: Empty list");      // IAE

   T result = itr.next();
   if (result == null)
      throw new NullPointerException("Min.min");                  // NPE

   while (itr.hasNext())
   {
      T comp = itr.next();
      if (comp.compareTo(result) < 0)
      {
         // for simplicity, assume no exception is thrown here
         result = comp;
      }
   }
   return result;
}
```

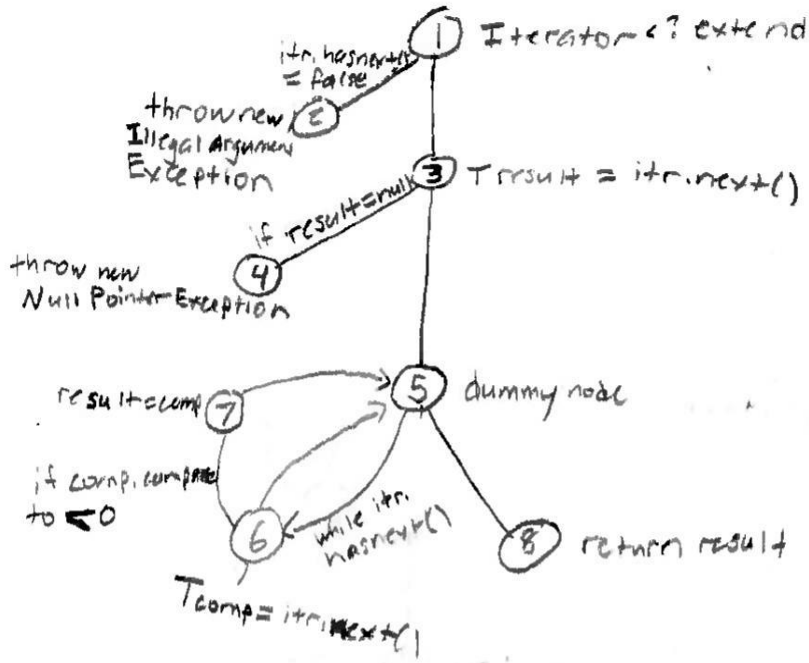
Note: `iterator()` returns an iterator that can be used to loop through a collection
`hasNext()` returns `true` if the iteration has more element(s)
`next()` returns the next element in the iteration

3/4

1. (4 pts.) Draw a **control flow graph** for this code. There are several possible values for the number of nodes in the graph. Choose something reasonable.

   (2 pts.) **Label edges/nodes** directly in the graph with the corresponding **code fragments**. Annotate only the code; no need to include defs and uses.

1.5/2



no initial node
-1

some edges unlabeled
-0.5

next time, bold final node circles

2. (4 pts.) Give **test cases (inputs and expected outputs)** that yield **edge coverage** on the graph. Hint: You should consider the number of test paths needed and then the test cases for the test paths.

2.5/4

T1: input{} out: Illegal argument exception

T2: input{0} out: Null pointer exception

T3: input{0.0} out: 0

T4:

-1 for not satisfy edge coverage

check case 2 input - 0.5

The score on this page is 7/10