**CS2030S**

**RECITATION 05**

Q1

a)

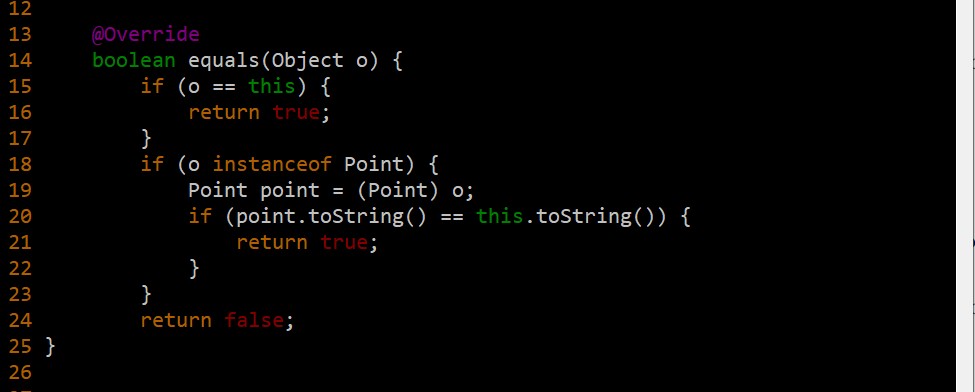
(1, 1) does not print this

(1, 1) does not print this

-1 only this comes out.

Its calling Objects ‘s equals, not Object ‘s equals

b)



First if-clause: compares points to see if the same chunk of memory is being compared

But why otherPoint.x can work tho? Since point declared x and y as private

c)

Object’s equals method has a hashcode, which is created when an object is instantiated. It generates an integer, which is more efficient than the equals method(Object c). so this HashSet’s equals is based on hashcode. ie HashSet does the comparison using the hashcode as opposed to using the Point’s equals method. Hashcode is consistent only in a single run of application, not necessarily across different runs

(1, 1)

(1, 1)

True (from p.equals(q))

True (from add(p))

True (from add(q))

[(1, 1), (1, 1)]

d) override the equals method in the point object with 2 parameters

wrong.

Have to Override the hashCode using some encryption algorithm (eg multiplication of primes etc)

Q2

1. list initialized => []

list.add(point1) => [point1]

list.add(point2) => [point1, point2] it checks that index 0 has an element already, then it proceeds to index 1 where it is vacant, and places the object inside at index 1]

Iterator<Point> iter = list.iterator();

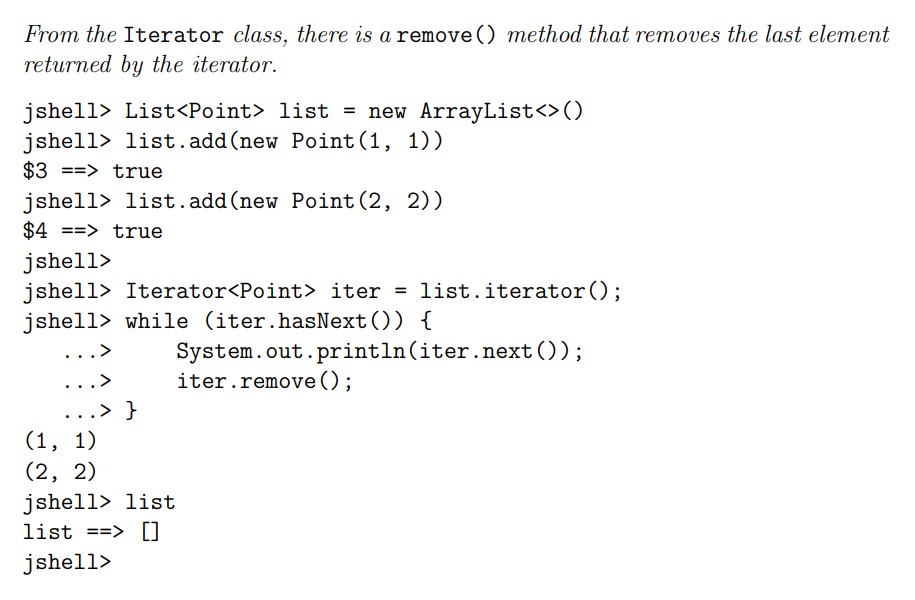
while (iter.hasNext()) {

System.out.println(iter.next());

}

1. iterator is generic but the for-loop is type-specific (?)

From the Iterator class, there is a remove() method that removes the last element returned by the iterator



Q3

214

main try running function g() which prints 2 then runs function f(), which throws an exception and the finally-clause prints 1. The f() throws an Exception to g(), which receives the exception and does not continue anymore, throwing the exception back to main. Main catches the exception and prints 4

read the ans key

Q4

read the ans key