**CT3535 Assignment B1**

*Source code:*

//Import ArrayList for infiniList

**import** java.util.ArrayList;

//Implements Runnable for Multi-threading

**public** **class** Iterator **implements** Runnable{

//infiniList, a static ArrayList of type String, to be filled with 3 strings

//"This", "is", & "infinite" infinitely

**public** **static** ArrayList<String> infiniList = **new** ArrayList<String>();

//Creating variables for code

**private** String name;

**private** Thread t;

//Iterator, which creates and calls the Thread instances

Iterator(String ThreadName){

name = ThreadName;

t = **new** Thread(**this**, name);

System.out.println(t + " Created");

t.start();

}

//Run function, which checks the last entry in the

//infiniList ArrayList, & adds the next string value into it

//then prints out the current state of the infiniList ArrayList

**Public synchronized** **void** run() {

//Posterity printout code to know if the program is working

System.out.println(t + "Running");

**try** {

**while**(**true**) {

//checks if 'This' was the last value, then adds 'if' if true

**if**(infiniList.get(infiniList.size()-1) == "This") {

infiniList.add("is");

System.out.println(infiniList);

}

//checks if 'is' was the last value, then adds 'infinite' if true

**else** **if**(infiniList.get(infiniList.size()-1) == "is") {

infiniList.add("infinite");

System.out.println(infiniList);

}

//adds infinite otherwise

**else** {

infiniList.add("This");

System.out.println(infiniList);

}

}

//a catch statement for any exceptions that the code throws up

//prints out that the thread interrupted, & what the problem is

}**catch** (Exception e) {

System.out.println(t + "Interrupted.");

System.out.println(e.getMessage());

}

//Posterity printout code to know if the program is working

System.out.println(t + "Exiting");

}

**public** **static** **void** main(String[] args) {

//adding This to the list to start the programs conditional code off

infiniList.add("This");

//Thread instance declaraton

Iterator t1 = **new** Iterator("Thread 1");

Iterator t2 = **new** Iterator("Thread 2");

Iterator t3 = **new** Iterator("Thread 3");

}

}

*Screenshot of code running:*

