**COMP 333**
**Fall 2019**

## Reference Counting and Garbage Collection

1.) Consider the function below:

```
function acyclic() {
  let obj1 = new Object();
  let obj2 = new Object();
  obj1.foo = obj2;
  return obj2;
}
```

1.a) For each line, write out what memory looks like after the line finishes executing if **reference counting** is used. Include the reference count in each object.

1.b.) Show what memory looks like after the function finishes executing if **garbage collection** is used. Assume that a GC cycle is triggered after the last line finishes executing.

2.) Consider the function below:

```
function cyclic() {
  let obj1 = new Object();
  let obj2 = new Object();
  obj1.foo = obj2;
  obj2.foo = obj1;
}
```

2.a) For each line, write out what memory looks like after the line finishes executing if **reference counting** is used. Include the reference count in each object.

2.b) You should see something strange with this memory diagram. What is strange?

2.b.) Show what memory looks like after the function finishes executing if **garbage collection** is used. Assume that a GC cycle is triggered after the last line finishes executing.