

Andrew McDonald Question 1

(a) import java.util.*;

class SetsWithOps implements Comparable, Comparator {

private ArrayList<Integer> list = new ArrayList();

```
public SetsWithOps(ArrayList<Integer> intList) {
    list.clear();
    for (int i = 0; i < intList.size(); i++) {
        list.add(intList.get(i));
    }
}
```

(c)

public SetsWithOps multOp(SetsWithOps A, SetsWithOps B) {

ArrayList<Integer> list = new ArrayList();

```
for (int i = 0; i < A.getList().size(); i++) {
    if (B.getList().contains(A.getList().get(i))) {
        list.add(A.getList().get(i));
    }
}
```

{

SetsWithOps temp = new SetsWithOps(list);
return temp;

}

AM

b)

```
public SetsWithOps appOp(SetsWithOps A, SetsWithOps  
ArrayList<Integer> list = new ArrayList();  
for (int i=0; i < A.getList().size(); i++) {  
    list.add(A.getList().get(i));  
}  
for (int k = 0; k < B.getList().size(); k++) {  
    list.add(B.getList().get(k));  
}  
SetsWithOps temp = new SetsWithOps(list);  
return temp;  
}
```

AM

(d)

```
public int compare(Object A, Object B) {
```

```
    SetsWithOps o = (SetsWithOps) A;
```

```
    SetsWithOps o2 = (SetsWithOps) B;
```

```
    int sum1 = 0
```

```
    int sum2 = 0
```

```
    for (int i = 0; i < o.getList().size(); i++) {
```

```
        sum1 += o.getList().get(i);
```

```
}
```

```
    for (int k = 0; k < o2.getList().size(); k++) {
```

```
        sum2 += o2.getList().get(k);
```

```
}
```

```
    if (sum1 < sum2)
```

```
        return -1;
```

```
    else if (sum1 == sum2)
```

```
        return 0;
```

```
    else
```

```
        return 1;
```

```
}
```

Continued on next page



AM

```
public int compareTo(Object A) {
    SetsWithOps o = (SetsWithOps) A
    int sum1 = 0;
    int sum2 = 0;

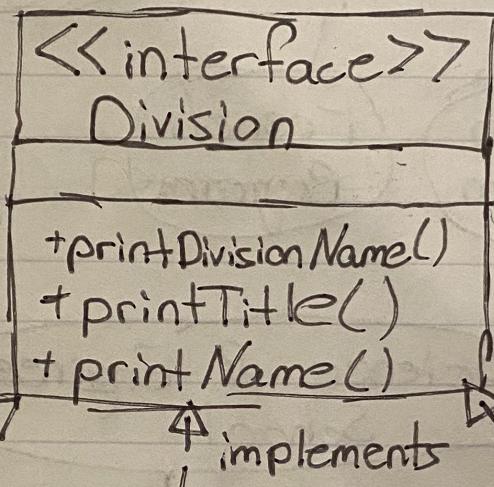
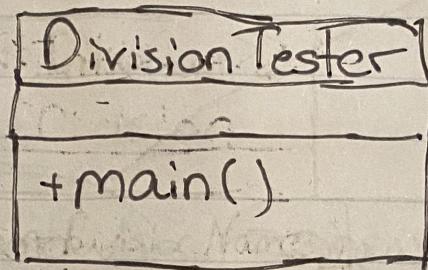
    for (int i = 0; i < this.list.size(); i++) {
        sum1 += this.list.get(i);
    }
    for (int k = 0; k < o.getList().size(); k++) {
        sum2 += o.getList().get(k);
    }

    if (sum1 < sum2)
        return -1;
    else if (sum1 == sum2)
        return 0
    else
        return 1
}
```

AM

Question 2

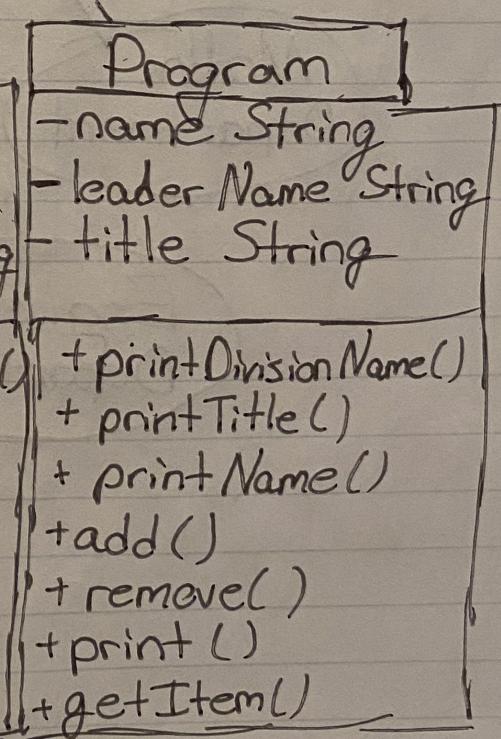
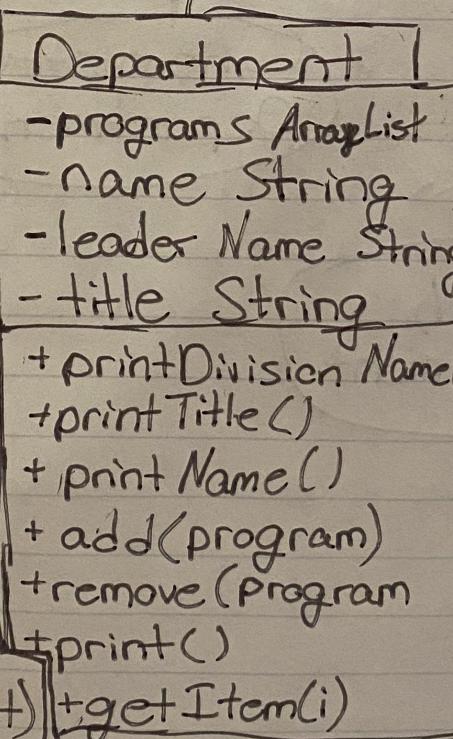
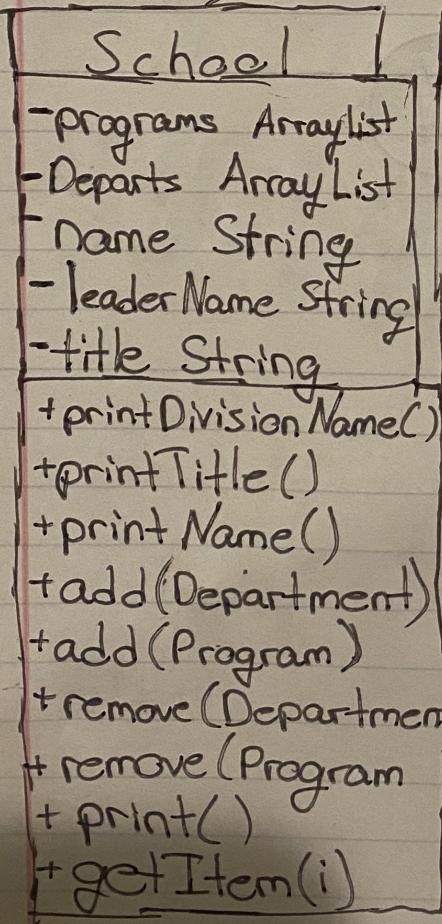
@



implements

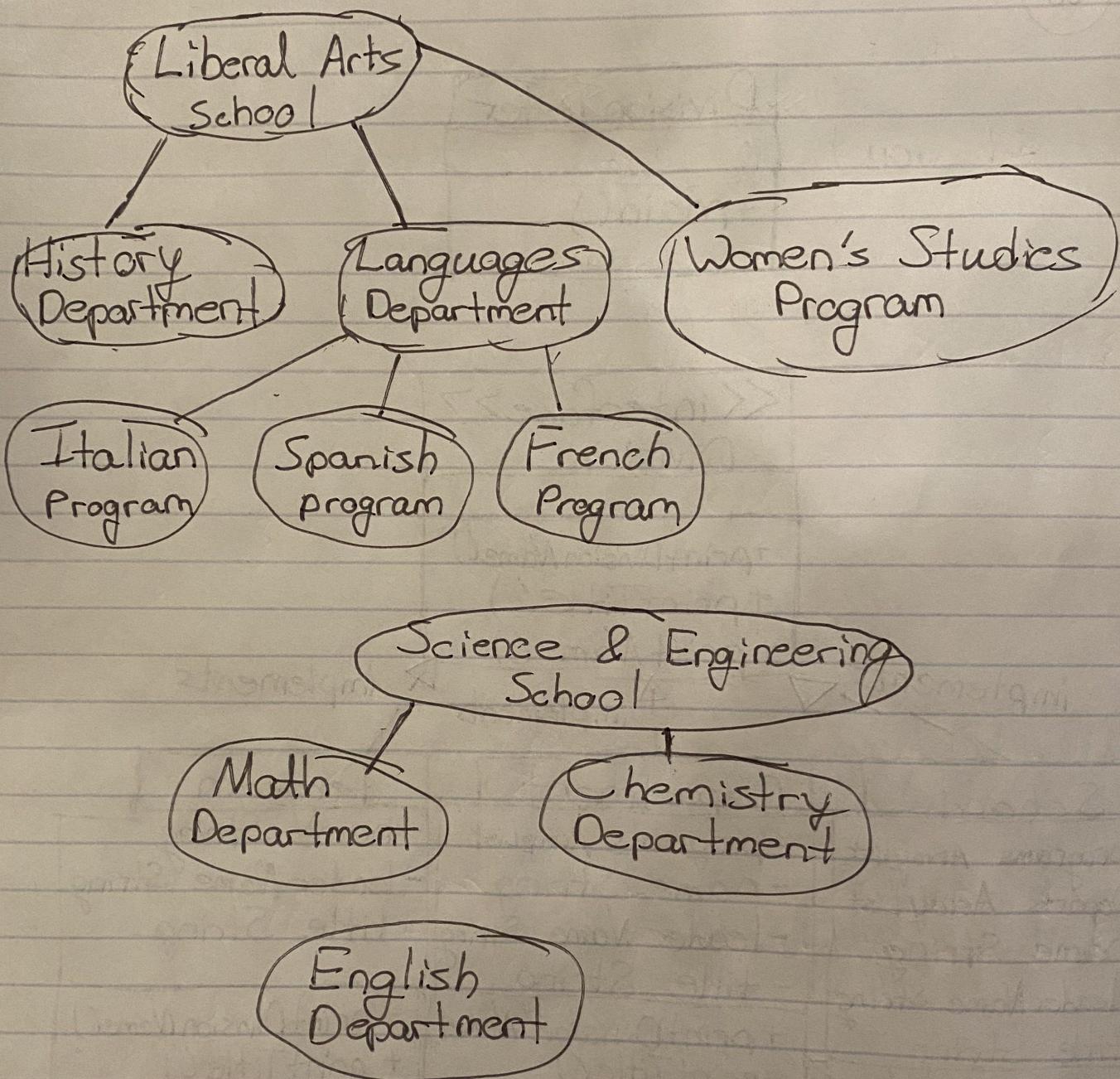
implements

implements



AM

(b)



AM

(C)

History
Language
Italian
Spanish
French
Women's Studies