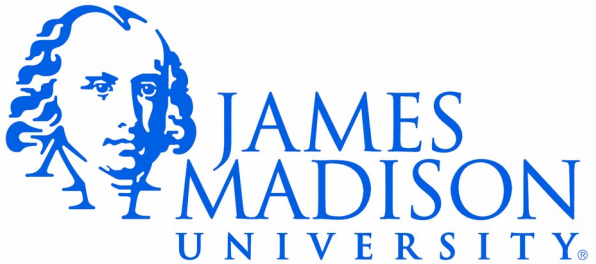
CS139 - Reference Arrays Review Activity



Exercise

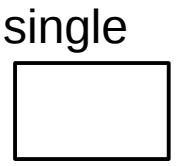
- Draw the memory diagram and determine output.

```
Die single;
Die[] dice1;
Die[] dice2;

dice1 = new Die[4];

single = new Die(1);

for (int i = 0; i < dice1.length; i++) {
   dice1[i] = single;
}

dice2 = dice1;

dice2[3] = new Die(4);

for (Die curDie : dice1) {
   System.out.println(curDie.getFace());
}
```

Exercise

- Draw the memory diagram and determine output.

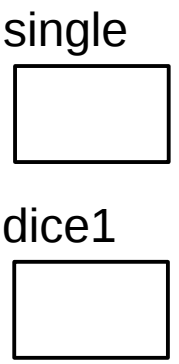
```
Die single;
Die[] dice1;
Die[] dice2;

dice1 = new Die[4];

single = new Die(1);

for (int i = 0; i < dice1.length; i++) {
   dice1[i] = single;
}

dice2 = dice1;

dice2[3] = new Die(4);

for (Die curDie : dice1) {
   System.out.println(curDie.getFace());
}
```



Exercise

- Draw the memory diagram and determine output.

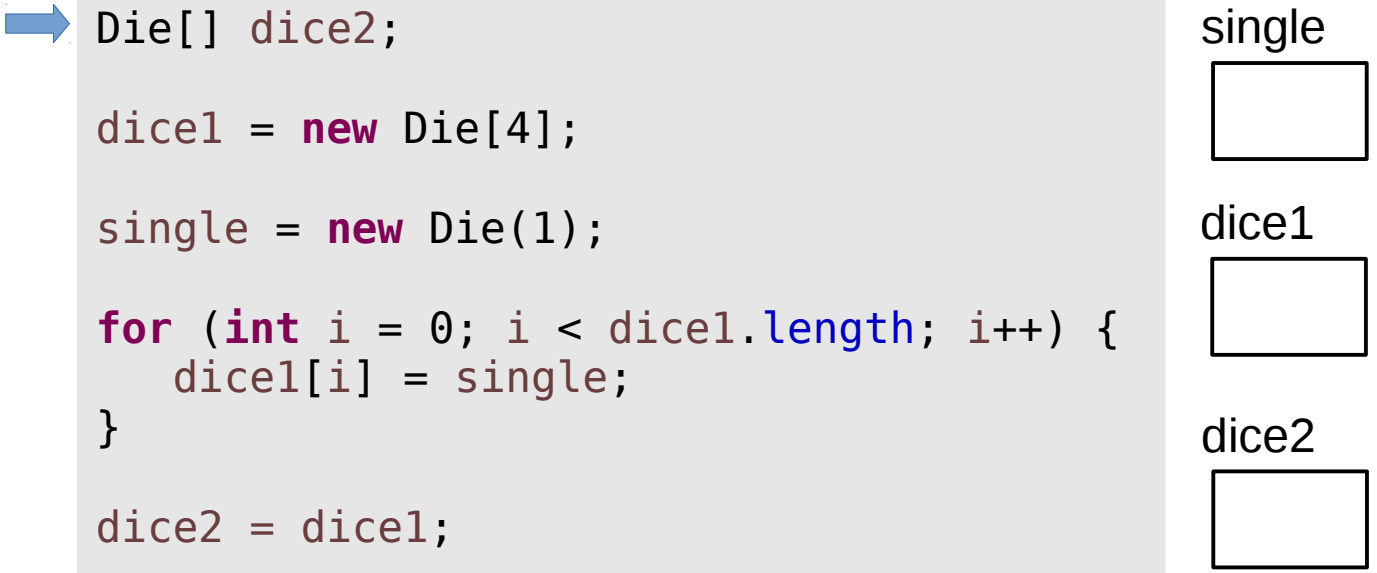
```
Die single;
Die[] dice1;
Die[] dice2;

dice1 = new Die[4];

single = new Die(1);

for (int i = 0; i < dice1.length; i++) {
   dice1[i] = single;
}

dice2 = dice1;

dice2[3] = new Die(4);

for (Die curDie : dice1) {
   System.out.println(curDie.getFace());
}
```



Exercise

- Draw the memory diagram and determine output.

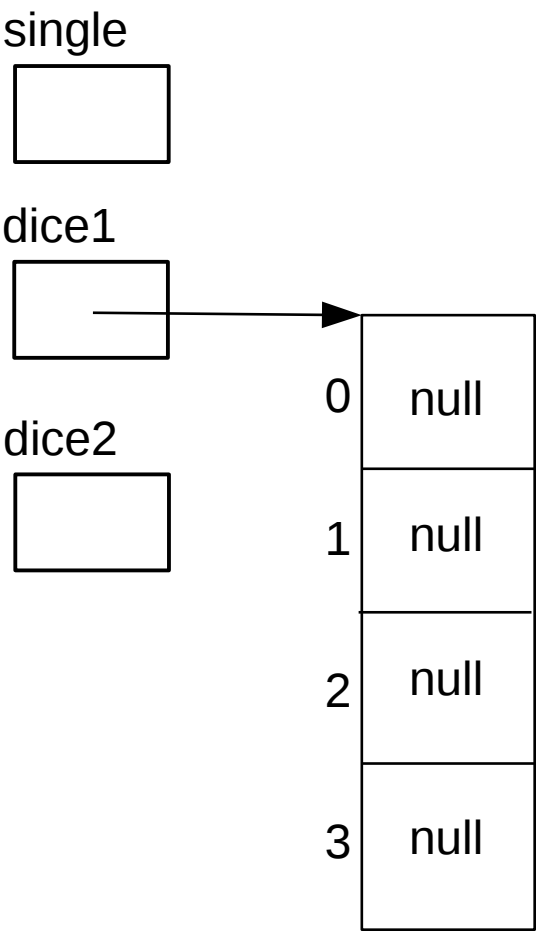
```
Die single;
Die[] dice1;
Die[] dice2;

dice1 = new Die[4];

single = new Die(1);

for (int i = 0; i < dice1.length; i++) {
   dice1[i] = single;
}

dice2 = dice1;

dice2[3] = new Die(4);

for (Die curDie : dice1) {
   System.out.println(curDie.getFace());
}
```



Exercise

- Draw the memory diagram and determine output.

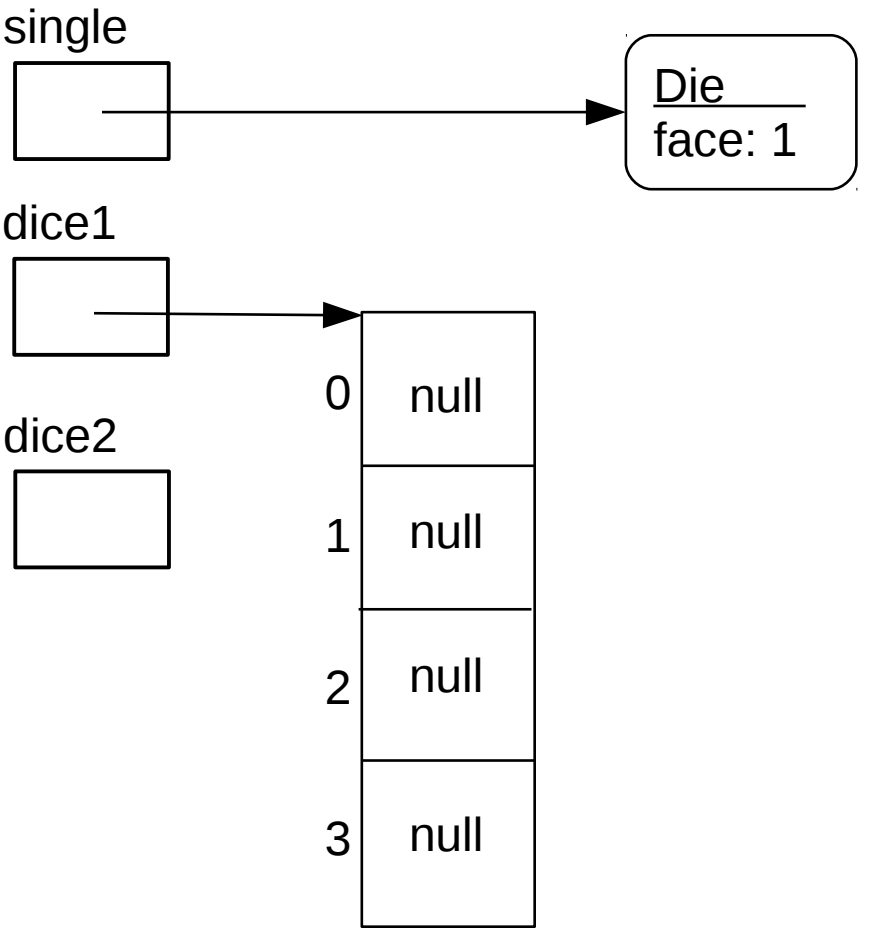
```
Die single;
Die[] dice1;
Die[] dice2;

dice1 = new Die[4];

single = new Die(1);

for (int i = 0; i < dice1.length; i++) {
   dice1[i] = single;
}

dice2 = dice1;

dice2[3] = new Die(4);

for (Die curDie : dice1) {
   System.out.println(curDie.getFace());
}
```



Exercise

- Draw the memory diagram and determine output.

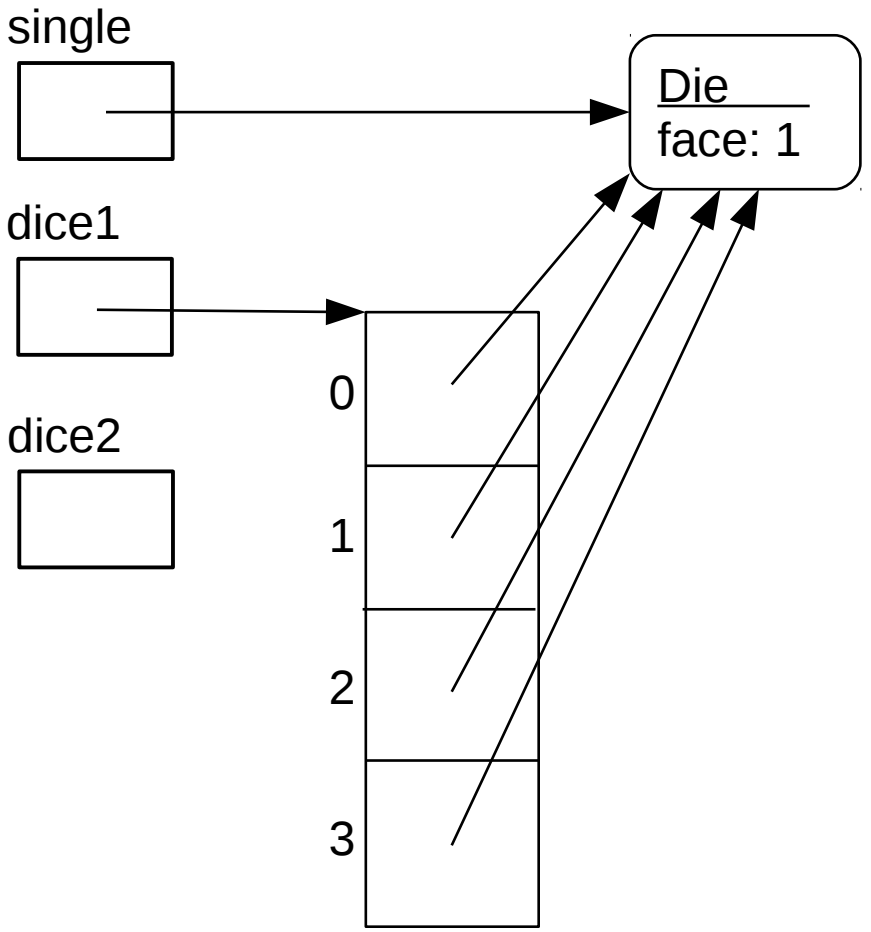
```
Die single;
Die[] dice1;
Die[] dice2;

dice1 = new Die[4];

single = new Die(1);

for (int i = 0; i < dice1.length; i++) {
   dice1[i] = single;
}

dice2 = dice1;

dice2[3] = new Die(4);

for (Die curDie : dice1) {
   System.out.println(curDie.getFace());
}
```



Exercise

- Draw the memory diagram and determine output.

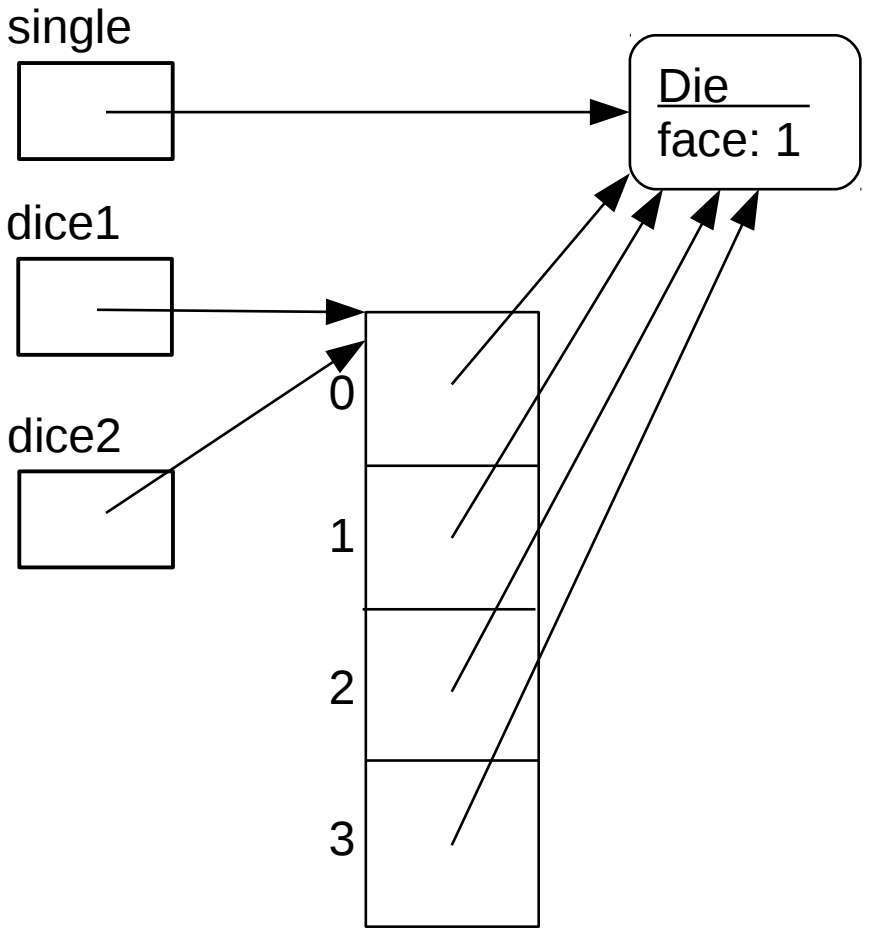
```
Die single;
Die[] dice1;
Die[] dice2;

dice1 = new Die[4];

single = new Die(1);

for (int i = 0; i < dice1.length; i++) {
   dice1[i] = single;
}

dice2 = dice1;

dice2[3] = new Die(4);

for (Die curDie : dice1) {
   System.out.println(curDie.getFace());
}
```



Exercise

- Draw the memory diagram and determine output.

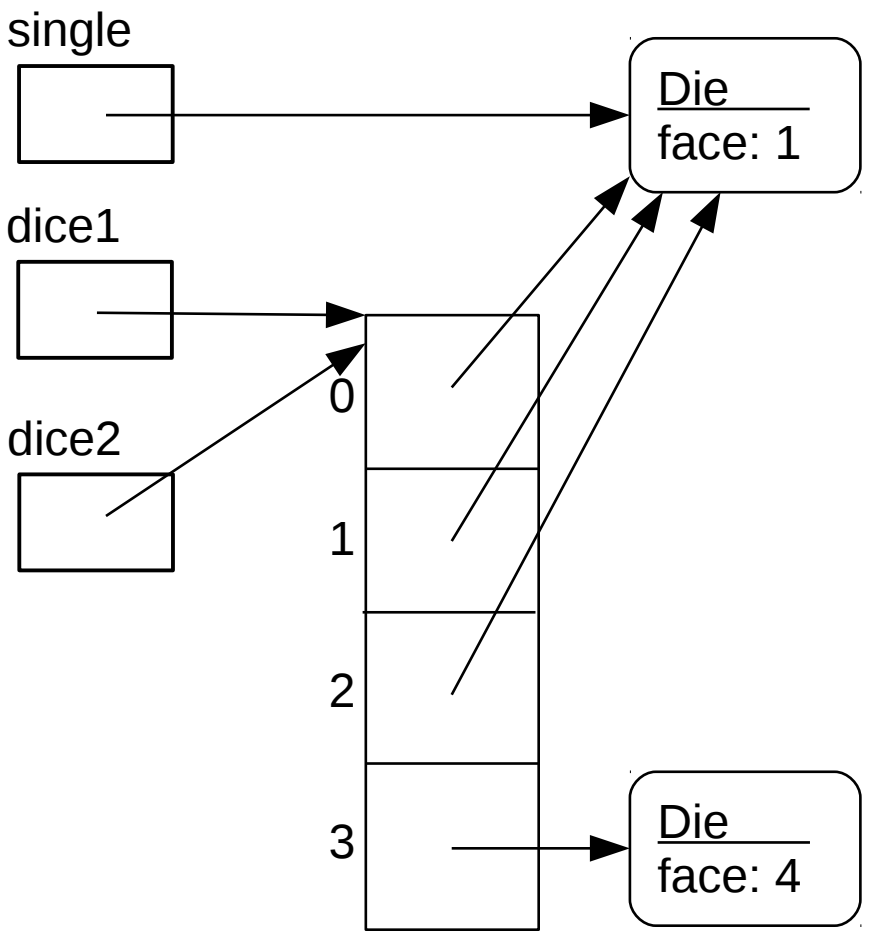
```
Die single;
Die[] dice1;
Die[] dice2;

dice1 = new Die[4];

single = new Die(1);

for (int i = 0; i < dice1.length; i++) {
   dice1[i] = single;
}

dice2 = dice1;

dice2[3] = new Die(4);

for (Die curDie : dice1) {
   System.out.println(curDie.getFace());
}
```



Exercise

- Draw the memory diagram and determine output.

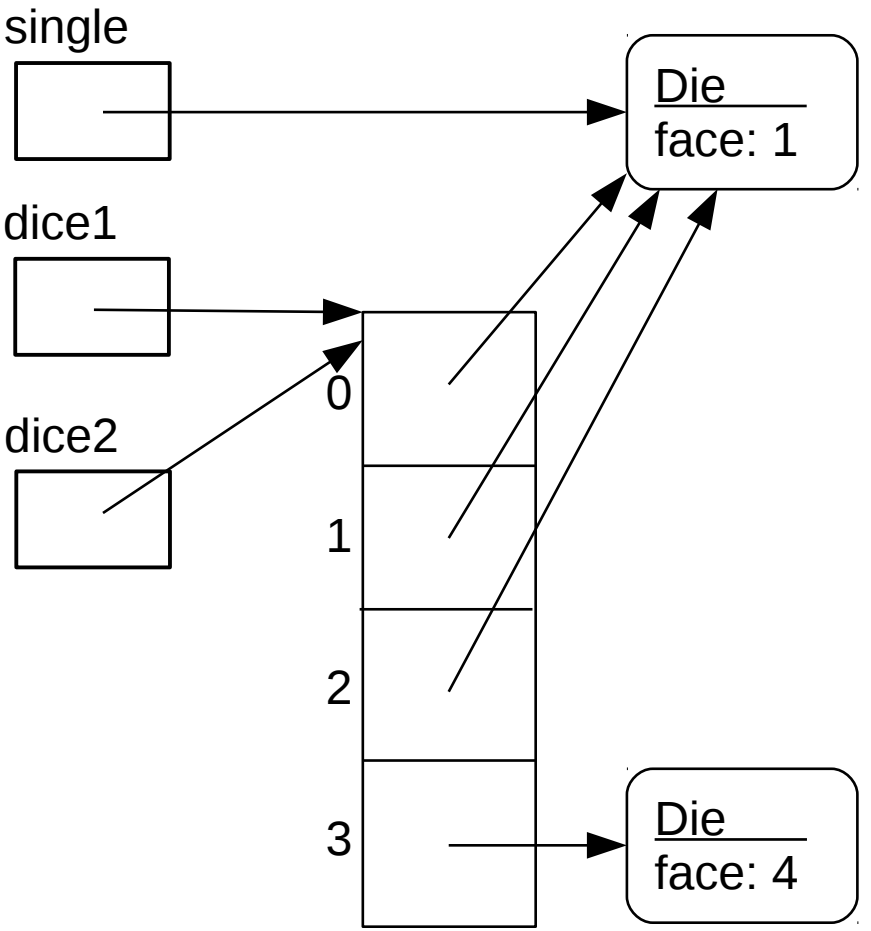
```
Die single;
Die[] dice1;
Die[] dice2;

dice1 = new Die[4];

single = new Die(1);

for (int i = 0; i < dice1.length; i++) {
   dice1[i] = single;
}

dice2 = dice1;

dice2[3] = new Die(4);

for (Die curDie : dice1) {
   System.out.println(curDie.getFace());
}
```



Exercise

- Draw the memory diagram and determine output.

```
Die single;
Die[] dice1;
Die[] dice2;

dice1 = new Die[4];

single = new Die(1);

for (int i = 0; i < dice1.length; i++) {
   dice1[i] = single;
}

dice2 = dice1;

dice2[3] = new Die(4);

for (Die curDie : dice1) {
   System.out.println(curDie.getFace());
}
```



Output:
1
1
1
4