Problem Set 2, Part I

Problem 1: String objects and their methods

1-1

```
a) s1.substring(6) + " " + s2.substring(0, 2)
b) s1.substring(6, 7) + s1.substring(9) + “ “ + s2.substring(5)
c) s1.toUpperCase().substring(0) + s1.toUpperCase().substring(9) +
s2.substring(9)
d) s1.substring(0, 1) + s1.substring(9) + s2.substring(0, 2)
e) s1.charAt(8)
f) s1.substring(8, 9)
g) s1.substring(0, 1) + s2.substring(0, 1)
h) s1.indexOf(“i”)
i) s1.replace(‘t’, ‘u’)
```

Problem 2: Understanding code that uses an array

2-1-1)

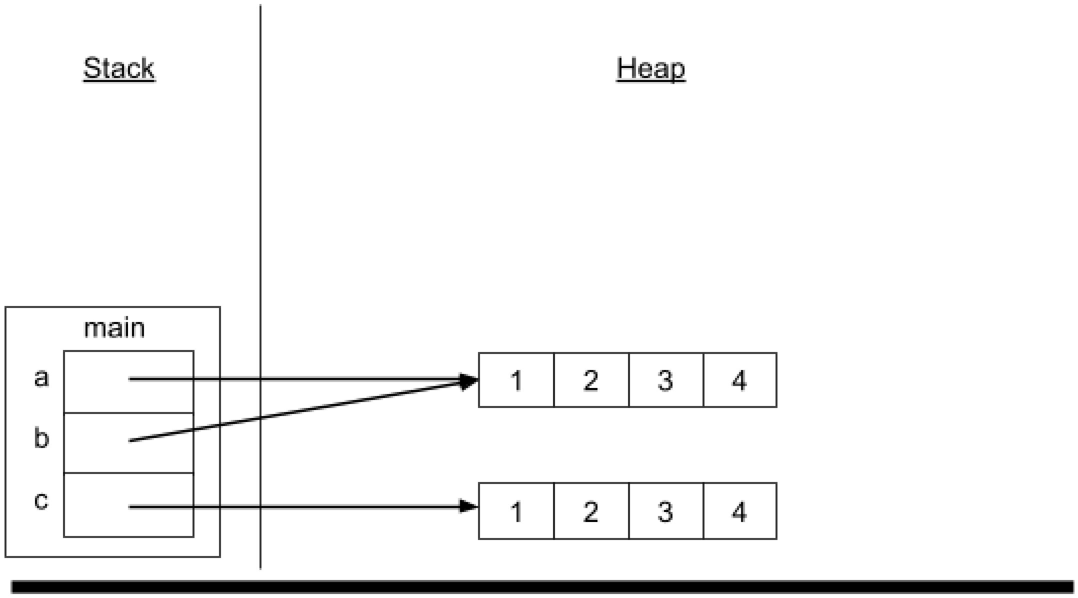
i	values
-	{0, 1, 2, 3, 4, 5, 6, 7}
1	{0, 0, 2, 3, 4, 5, 6, 7}
3	{0, 0, 2, 2, 4, 5, 6, 7}
5	{0, 0, 2, 2, 4, 4, 6, 7}
7	{0, 0, 2, 2, 4, 4, 6, 6}

2-2) {6, 0, 2, 2, 4, 4, 6, 6}

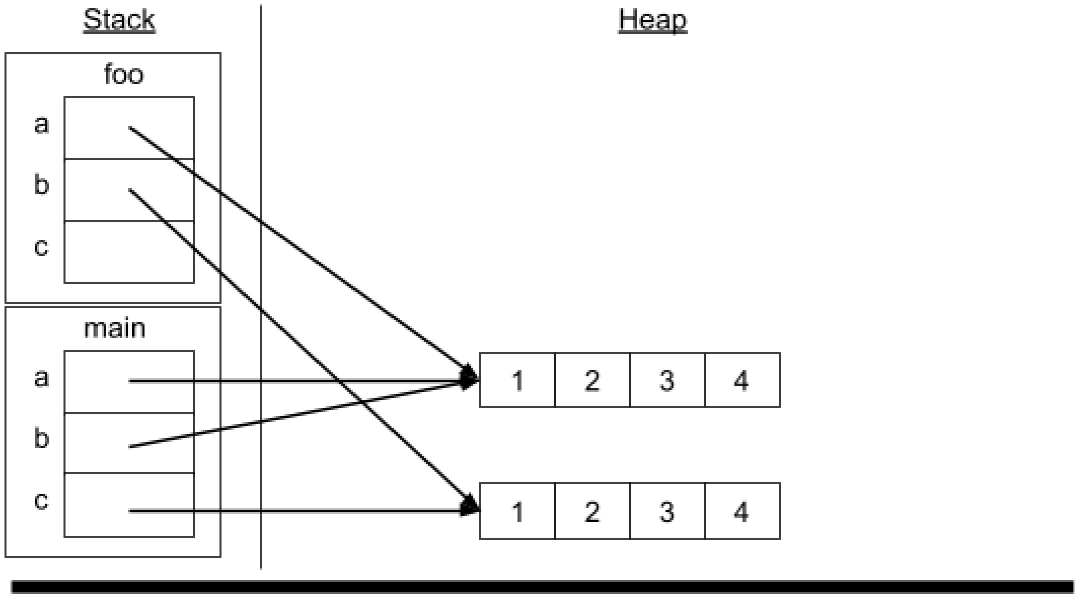
2-3) We will see the changes made by the call to the mystery method. The array is passed by reference to methods, meaning that the method “mystery” operates on the original array object, thus modifying its contents directly.

Problem 3: Memory management and arrays

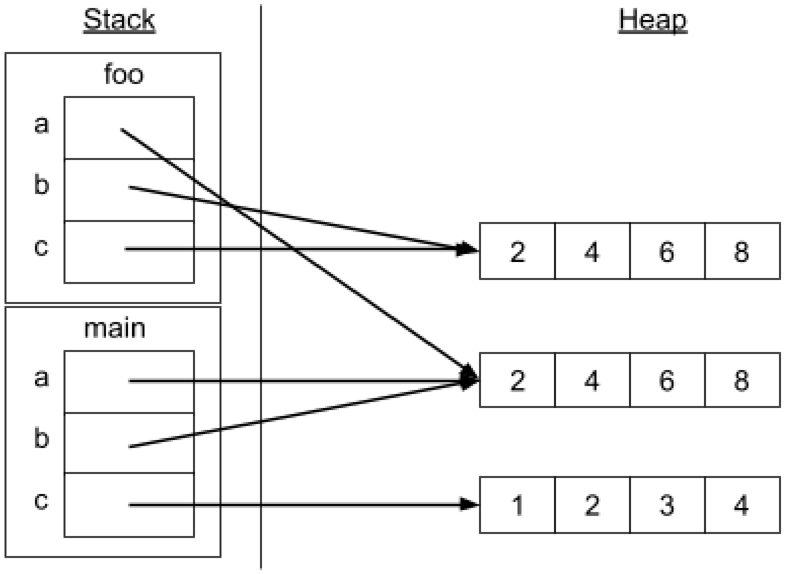
3-1)



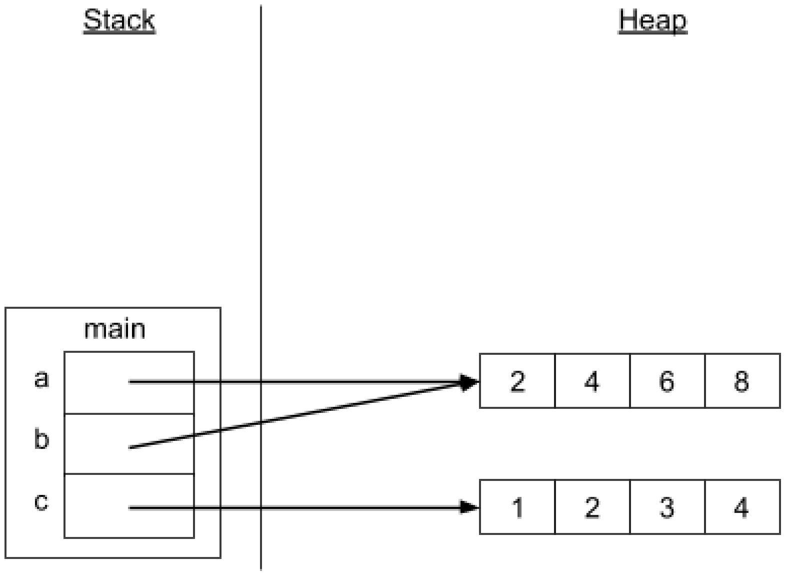
3-2)



3-3)


Stack
Heap
foo
a
b
c
main
a
b
c
2 4 6 8
2 4 6 8
1 2 3 4

3-4)

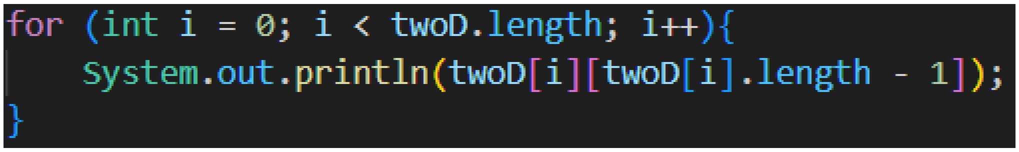

Stack
Heap
main
a
b
c
2 4 6 8
1 2 3 4

Problem 4: Two-dimensional arrays

4-1)

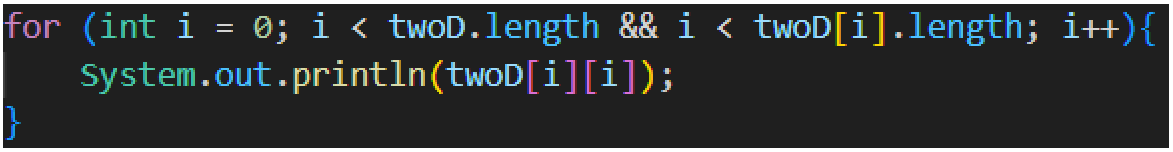
twoD[2][1] = 30;

4-2)



```
for (int i = 0; i < twoD.length; i++){
    System.out.println(twoD[i][twoD[i].length - 1]);
}
```

4-3)



```
for (int i = 0; i < twoD.length && i < twoD[i].length; i++){
    System.out.println(twoD[i][i]);
}
```

Problem 5: Our `Rectangle` class revisited

5-1)

type of method: mutator
header: public void shrink(int amount)

5-2)

type of method: accessor
header: public double diagonal()

5-3)

problems in code: The code attempts to access and modify the width and height attributes of rect. Since the Rectangle class uses encapsulation, those fields are private and inaccessible outside the class.

rewritten version:

```
Rectangle rect = new Rectangle(10, 20);
System.out.println(“width = “ + rect.width);
rect.grow(20, 0);
System.out.println(rect);
```