CS 61B
Spring 2019

Extra Practice Worksheet
Discussion 3: Objects and Arrays

Worksheet 3

1 Loading and Riding

Given the following lines of code, comment out the lines which are examples of overloading, overriding, or errors, and specify each one.

Solution:

```
class Library{
    int add(int a,int b)
    float add(float a,int b) //overloading
    float add(int a ,float b) //overloading
    void add(float a) //overloading
    int add(int a) /overloading
    void add(int a) //error conflict with the  method int add(int a)
}
class BookDetails {
  String title;
  setBook(String title){}
}
class ScienceBook extends BookDetails {
  setBook(String title){} //overriding
  setBook(String title, String publisher,float price){} //overloading
}
```

2 Reversing Arrays

1. Write code to reverse an array *iteratively*

```
public static void reverse (int [] arr) {
    for(int i = 0, i < ________________, i++) {
        int temp = ___________________________;
        arr[____________________] = arr[___________________________];
        arr[___________________________] = ___________________________;
    }
}
```

Solution:

```
public static void reverse (int [] arr) {
    for(int i = 0, i < arr.length / 2, i++) {
```

```
        int temp = arr[i];
        arr[i] = arr[arr.length-i-1];
        arr[arr.length-i-1] = temp;
    }
}
```

2. Write code to reverse an array *recursively*

```
public static void reverse (int [] arr) {
    reverseHelper(________, 0, ________________________)
}
static void reverseHelper(int[] arr, int beg, int end){
    if(________________________){
        int temp = ________________________;
        arr[________________________] = arr[____________________];
        arr[________________________] = ________________________;
        ________________________________;
        ________________________________;
        ____________________________________________________;
    }
}
```

Solution:

```
public static void reverse (int [] arr) {
    reverseHelper(arr, 0, arr.length - 1)
}
static void reverseHelper(int[] arr, int beg, int end){
    if(beg < end){
        int temp = arr[beg];
        arr[beg] = arr[end];
        arr[end] = temp;
        beg++;
        end--;
        reverseHelper(arr, beg, end);
    }
}
```

3 Confusing Constructors

What is the output of the following program after we execute the main method?

```
public class Confusing {
    private Confusing(Object o) {
        System.out.println("Data Structures");
    }
    private Confusing(double[] dArray) {
        System.out.println("Algorithms");
    }
    public static void main(String[] args) {
        double[] array = new double[4];
        IntList list = IntList.list(array);
        Confusing Ell = new Confusing(array);
        Confusing Mow = new Confusing(list);
        Confusing Shub = new Confusing(null);
    }
}
```

Solution:

Algorithms

Data Structures

Algorithms

When we we make the call `Confusing Shub = new Confusing(null)`

Java recognizes that both constructor methods are capable of carrying out the construction of a Confusing object. The constructor Confusing(Object) accepts any parameter passed to Confusing(double[]), so Confusing(Object) is less specific. (Every double array is an Object, but not every Object is a double array). The compiler, therefore, selects the most "specific" overloading chooses the second constructor.