CSCI 161 Final Exam - STUDY GUIDE

Know the following terms

abstraction
accessor
array
auto-initialization
behavior
client code
constructor
encapsulation
implicit parameter
index
instance method
jagged array
mutator
object
object-oriented programming
private fields
reference semantics
state
this
value semantics
zero-based indexing

Know these topics and be able to answer questions related:

1) Arrays and how they differ from primitive data type variables and when they are useful.

2) The differences between Reference and Value Semantics.

3) The differences between procedural and object-oriented programming.

4) Definitions of class and object and how they are related.

5) The limitations of Java arrays, including no ability to resize (cannot grow as needed) and how to work around those issues (especially when reading files into arrays).

6) The “gotchas” of file I/O, including reading past end of file, trying to read the wrong data types at the wrong time and how to avoid these things.

Program Tracing*:*

Be able to trace code like the following, answering questions like the following:

7) What are the values of the elements in the array `numbers` after the following code is executed:

```
int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
for (int i = 0; i < 5; i++) {
    numbers[i] = numbers[i + 5];
}
```

8) What elements does the array `list` contain after the following code is executed:

```
int[] list = {2, 8, 18, -4, 6, 12};
for (int i = 0; i < list.length; i++) {
   list[i] = list[i] + (list[i] / list[0]);
}
```

9) Given **the** following file contents, (5 lines), what will the output be for each of the code fragments that follow (*use back of sheet if needed*):

```
Now is the time
      for all good
   men to

come to the aid of their party
```

a.

```
Scanner input = new Scanner(new File(“now.txt”));
while (input.hasNextLine()) {
   String line = input.nextLine();
   System.out.println(line);
}
```

b.

```
Scanner input = new Scanner(new File(“now.txt”));
while (input.hasNext()) {
   String token = input.next();
   System.out.println(token);
}
```

Programming*:*

Be able to write code for problems like the following:

10) Write code that uses a **for** loop to store all odd numbers between -12 and 22 in an array using a loop. Make the array's size *exactly* large enough to store the numbers.

11) Write code that uses a **for** loop to print each element of an array named data that contains six integers, populating your array as part of its declaration and initialization. Your code should generate the following output, *exactly*:

```
element [0] is 1
element [1] is 2
element [2] is 34
element [3] is -3
element [4] is 259
element [5] is -88
```

12)Write a method called `sortPairs` that accepts an array of integers and sorts the elements at adjacent indexes. That is, element 0 and 1 are sorted, element 2 and 3 are sorted, and so on. If the array has an odd length the final element should be left unchanged. For example, the list [10, 5, 20, 8, 11] should become [5, 10, 8, 20, 11].

13)Assume that a two-dimensional, rectangular array of integers called `data` has been declared with five rows and six columns. Write a loop to initialize the second row to store the numbers 6 through 1 in descending order.