DON'T WRITE ANYTHING HERE!!!

Name: (as it would appear on official course roster)	Seat:
Umail address: @umail.ucsb.edu	

1

EXAM: e02: Midterm 2

e02

CS56 f18

ready?	date	points
true	Mon 11/19 12:30PM	100

You may not collaborate on this exam with anyone. If you need to use the restroom, you must leave your cell phone with the exam proctor before leaving the room.

- Write your name at the top of this page **AND EVERY ODD NUMBERED PAGE.**
- Double check that you turned in ALL pages; look for "End of Exam" on the last page.
- This exam is **closed book, closed notes**, **closed mouth, cell phone off.**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes.
- This sheet will be collected with the exam, and might not be returned.
- Please write your name on your notes sheet.

---

1. When working with Java code, we sometimes download `.jar` files, and sometimes create `.jar` files.

   a. (6 pts) What is a jar file?

   b. (6 pts) To make a jar file executable, you have to specify one piece of information that goes into the so-called **manifest** of the jar file. What is this information?

2

e02

CS56 f18

2. For this problem, you may find it helpful to consult the reference material regarding interfaces and methods related to sorting on Handout B.

- Below, you will code for the file `Student.java`. This code compiles cleanly as given.
- However, as a result of something missing in `Student.java`, the code for `StudentMain.java` shown on Handout A, does not compile; it gives an error message as shown.

Suppose you were shown this code at a job interview, and asked to explain in plain english, why the code doesn't compile, and to fix the code. The interviewer gives you two hints:

a. There is **nothing wrong** with `StudentMain.java`; rather the problem is that two changes need to be made in `Student.java`

b. Make sure you pay attention to the comment on line 8 of `StudentMain.java`

a. (16 pts) Make the two changes needed, directly on the listing below.
You may cross out code, and/or write new code where it belongs.

```
1  public class Student {
2
3
4      private String name;
5      private int perm;
6      private String major;
7
8
9      public Student (String name, int perm, String major) {
10         this.name = name;
11         this.perm = perm;
12         this.major = major;
13     }
14
15
16     public String getName() { return name; }
17     public int getPerm() { return perm; }
18     public String getMajor() { return major; }
19
20
21
22
23
24
25
26
27
28 }
```

b. (8 pts) How would you explain your fixes to the interviewer? Write in plain english at an appropriate level of abstraction; so that the interviewer has confidence in your technical knowledge.

DON'T WRITE ANYTHING HERE!!!

Name: (as it would appear on official course roster)	Seat:
Umail address: @umail.ucsb.edu	

3 e02

CS56 f18

3. Exceptions in Java can be divided into two broad categories:

   - One category is the kind that, if there is any chance it can happen in the code has to be "caught or declared to be thrown"
   - The other category is the kind that can happen, but doesn't have to be "caught or declared to be thrown".

a. Consider the kind that DOES have to be "caught, or declared to be thrown". Indicate for each statement, whether the statement is true or false, by checking the appropriate box.

(2 pts)	It is a subclass of `RuntimeException`	☐ True	☐ False
(2 pts)	If it is thrown, there is a problem with the program logic	☐ True	☐ False
(2 pts)	`IndexOutOfBoundsException` would be an example of this type of exception	☐ True	☐ False
(2 pts)	`FileNotFoundException` would be an example of this type of exception	☐ True	☐ False

b. Consider the kind that does NOT have to be "caught, or declared to be thrown". Indicate for each statement, whether the statement is true or false, by checking the appropriate box.

(2 pts)	It is a subclass of `RuntimeException`	☐ True	☐ False
(2 pts)	If it is thrown, there is a problem with the program logic	☐ True	☐ False
(2 pts)	`UnknownHostException` (often a symptom of having no internet connection) would be an example of this type of exception	☐ True	☐ False
(2 pts)	`NullPointerException` would be an example of this type of exception	☐ True	☐ False

4

e02

CS56 f18

4. The class `java.lang.String` (i.e plain old java `String` objects) has a method:

`public int compareTo(String anotherString)`	Compares two strings lexicographically.
`public int length()`	Returns the length of this string

Note that perhaps contrary to expectations, `String` does NOT have a method `static int compare(String s1, String s2);`

Consider the `main` program in `StringSort.java` on Handout A which sorts some strings. Note that the place where the sort is supposed to appear, at lines 14-19 shows where one or more lines of code could go to sort the array.

You are now asked to fill in two different things that could go in that space. In each case, it is just one call to either `java.util.Collections.sort` or the `sort1` method of `ArrayList<String>` that does the sorting, but you may need extra space for a lambda expression.

For this problem, you may find it helpful to consult the reference material regarding interfaces and methods related to sorting on Handout B.

a. (10 pts) Sort the strings in lexicographic order, i.e. so that the output of the program is:

```
[a, an, be, cat, cow, dog, duck, goose, moose, pig, to]
```

b. (10 pts) Sort the first by their length, then in lexicographic (i.e. the order used by the `compareTo` method of `java.lang.String`).

That is, the output of the program should be:

```
["a","an","be","to","cat","cow","dog","pig","duck","goose","moose"]
```

Your answer should use a lambda expressions for an instance of `java.util.Comparator<String>`.

DON'T WRITE ANYTHING HERE!!!

Name: (as it would appear on official course roster)	Seat:
Umail address: @umail.ucsb.edu	

5

e02

CS56 f18

5. Chapter 1 of HFDP covered the strategy pattern, using an example of a Duck class, where you could set the FlyBehavior or a QuackBehavior at runtime. The strategy pattern, according to our textbook: "defines a family of algorithms, encapsulates each one, and makes them interchangeable. Strategy lets the algorithm vary independently from clients that use it."

   Consider the interfaces `java.lang.Comparable`, and `java.util.Comparator` and how they are used.

   a. (6 pts) Which of these is an example of the Strategy pattern? (check one)

   ☐ `Comparable` ☐ `Comparator` ☐ both ☐ neither

   b. (6 pts) Explain your answer to the multiple choice question above.

6

e02

CS56 f18

6. (6 pts) Some Java files start with lines of code such as this one:

```
package edu.ucsb.cs56.pluto;
```

What purpose do these lines serve? That is, what problem do they solve, and how do they solve it?

7. When we did the user story mapping exercise, we tried to work towards a "thin horizontal slice" to find a "minimum viable product".

a. (6 pts) What do we mean by "minimum viable product"?

b. (4 pts) The "thin horizontal slice" is a slice across the story map from left to right. What does this slice represent? That is, what are the vertical columns that we are slicing through?

End of Exam