

CMSC131

Final Exam

First Name: _____

Last Name: _____

University ID: _____

Grader Use Only:

| | | |
|--------------|--|--------------|
| Page 2 | | (8) |
| Page 3 | | (10) |
| Page 4 | | (7) |
| Page 5 | | (8) |
| Page 6 | | (8) |
| Page 7 | | (12) |
| Page 8 | | (10) |
| Page 9 | | (8) |
| Page 10 | | (21) |
| Page 12 | | (8) |
| Total | | (100) |

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Your signature:

General Rules:

- This exam is closed-book and closed-notes.
- If you have a question, please raise your hand.
- Total point value is 100 points
- The duration of the exam is 2 hours.

1. (1 point) How many bytes are there in a gigabyte?
2. (1 point) How many different values can be represented with n bits?
3. (1 points) How many bytes of memory are required to store a variable of type `short`?
4. (1 point) When your Java program is compiled, what kind of file is created?
5. (2 points) Write the binary representation of the number twenty-nine.
6. (1 point) Which of the following are valid **Java identifiers** that could be used to name methods or variables? (Circle all the valid ones.)

`the*end``int``For``25years``pizza77`
7. (1 point) List the following operators in order of precedence. (Put the operator with highest precedence at the top; lowest at the bottom.)

List the operators here:

`+=` _____

`||` _____

`<=` _____

`&&` _____

`!` _____

8. (3 points) Assume that there is an interface called `Whimsical` that is implemented by a class called `Comedian`. Which of the following are valid?

- | | | |
|--|--------------|----------------|
| a. <code>Whimsical x = new Comedian();</code> | VALID | INVALID |
| b. <code>Comedian y = new Whimsical();</code> | VALID | INVALID |
| c. <code>Whimsical[] z = new Whimsical[10];</code> | VALID | INVALID |

9. (1 point) What key word can you put in front of a variable declaration to prevent the variable's value from changing during program execution?

10. (2 points) Consider a class called `Kennel` with a single instance variable representing an array of `Dog` references. **You are writing a "getter" for the array, but you don't want to allow any kind of modification to the state of the Kennel, and you don't want to make unnecessary copies.**

- a. If `Dog` is an *immutable* class, you should return a reference to what? (Circle one.)
the array shallow copy of the array deep copy of the array
- b. If `Dog` is a *mutable* class, you should return a reference to what? (Circle one.)
the array shallow copy of the array deep copy of the array

11. (1 point) When you pass a primitive variable as an argument to a method, it is possible for the method to modify the value of this variable. **TRUE / FALSE**

12. (3 points) Your friend has written a method called `onePercent`, which should return 1% of a given double value. He has written many JUnit tests for the method, including the one below.

```
@Test
public void test() {
    for (int i = 0; i < 1000000; i++) {
        double value = Math.random(); // generates random double between 0 and 1
        assertTrue(onePercent(value) == value * 0.01);
    }
}
```

- a. What do you think of this test? (Circle one.)

Useful test Useless test

- b. Why?

13. (2 points) Suppose you are testing an algorithm you have implemented on various sets of data, and the running times (measured with a stopwatch) appear in the table below.

| Size of dataset | Time (milliseconds) |
|-----------------|---------------------|
| 1000 | 300 |
| 2000 | 321 |
| 4000 | 343 |
| 8000 | 364 |
| 16000 | 384 |

What is your best guess as to the running time for the algorithm? (Circle one.)

$O(1)$ $O(\log n)$ $O(n)$ $O(n \log n)$ $O(n^2)$ $O(2^n)$

14. (2 points) Consider the array created below:

```
Object[] myList = new Object[10];
```

What kinds of things could be put into this array? (Circle all that apply.)

String references ints Object references Scanner references

15. (2 points) Name two methods that every class inherits from the Object class.

16. (1 point) What is the name of the IDE that we used in this course?

17. (3 points) Assume there is a class called `Pizza` in a package called `food`.

- a. What is the first statement at the top of the file `Pizza.java`?
- b. If you are writing a class that will make use of `Pizzas` from time to time, what might you put at the top of your class so that you can avoid typing out the fully qualified name repeatedly?
- c. Show how you would declare an instance variable of the `Pizza` class (an `int` called `radius`) so that it has *package visibility*. (Just write the variable declaration.)

18. (1 point) A class can directly implement more than one interface. TRUE / FALSE

19. (1 point) A class can directly extend more than one other class TRUE / FALSE

20. (1 point) Which of the following types can be used with switch statements? (Circle all that apply.)

`int`

`long`

`String`

`char`

21. (2 points) Rewrite the code fragment below as *a single statement* in which the keyword `return` appears only once:

```
if (x > 20) {  
    return 55;  
} else {  
    return 0;  
}
```

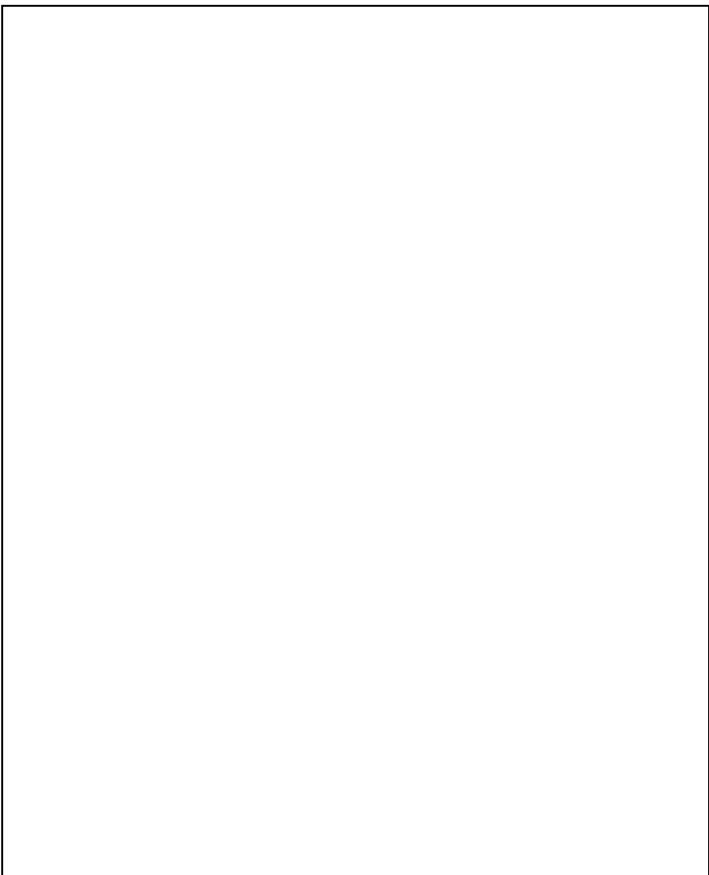
22. (8 points) Draw a memory diagram showing both the stack and the heap at the moment this program reaches the point marked **/* HERE */**. Be sure to use the same style that was demonstrated in class many times.

```
public class MemoryMap {  
  
    public static void process(String[] persons, int value) {  
        value += 200;  
        String selected = persons[1];  
        persons[1] = "Linda";  
        persons = null;  
        /* HERE */  
    }  
  
    public static void main(String[] args) {  
        String[] members = {"Jose", "Peter", "Mary"};  
        int max = 100;  
        process(members, max);  
    }  
}
```

Call Stack



Heap



23. (12 points) Specify the output for each of the following code fragments. Write your answers in the space to the right.

Write output here:

| | |
|--|--|
| a. <code>System.out.println(11 / 4);</code> | |
| b. <code>System.out.println(39 % 6);</code> | |
| c. <code>String x = "cow"; String y = new String(x) System.out.println(x.compareTo(y)); System.out.println((x == y)? 2 : 7); if (x.equals(y)) { System.out.println("YES"); } else { System.out.println("NO"); }</code> | |
| d. <code>int x = 3; System.out.println(2 + x++); System.out.println(13 - ++x); System.out.println(x);</code> | |
| e. <code>int y = 4; if (y > 2 y++ == 5) { System.out.println("OK"); } System.out.println(y);</code> | |
| f. <code>for (int i = 0; i < 100; i += 10) { if (i < 50) { continue; } if (i == 60) { break; } i -= 5; System.out.println(i); }</code> | |

24. (10 points) Fill in the method below using a *recursive* implementation. The method will return the smallest value in the array. You may assume that the array is not empty. You may include a helper method if you find it useful. Be sure your algorithm is efficient (not wasting time or memory).

```
public static double findSmallest(double[] a) {
```


25. (8 points) Assume there is a static method called `answerAnyQuestion` that takes a `String` parameter. Its prototype is below:

```
public static String answerAnyQuestion(String question)
```

This magical method will return the answer to *any* question, but... it will throw exceptions in the following cases:

- `NullPointerException` is thrown if you pass it "null" instead of an actual question
- `TooEasyException` is thrown if the length of the question is less than 10 characters
- `AreYouCrazyThatIsWayTooHardException` is thrown if the length of the question is longer than 30 characters

Write the output for the following program in the box at the bottom of the page:

```
private static void askIt(String question) {
    try {
        String response = answerAnyQuestion(question);
        System.out.println("A");
    } catch (NullPointerException e) {
        System.out.println("B");
    } catch (TooEasyException e) {
        System.out.println("C");
    } finally {
        System.out.println("D");
    }
}

public static void main(String[] args) {
    try {
        askIt("How do you do Problem 27?");
        askIt("Hint?");
        askIt("Could you please just tell me some PART of the correct answer?");
        System.out.println("E");
    } catch (TooEasyException e) {
        System.out.println("F");
    } catch (AreYouCrazyThatIsWayTooHardException e) {
        System.out.println("G");
    } finally {
        System.out.println("H");
    }
    System.out.println("I");
}
```

OUTPUT:

26. (21 points)

Recall the following instance methods of the `ArrayList<E>` class:

```
boolean add(E element)           // adds the element to the end of the list
int size()                       // returns the size of the list
boolean contains (Object element) // returns true if the list contains the element, false otherwise
```

Write a class called `WordSet` with the following features. **Any loops you write for this problem must be for-each loops.** (If you run out of space, you may continue on the next page.)

- A private instance variable called `words`, which is an `ArrayList` of `StringBuffer` references.
- A constructor that takes no parameters, and initializes `words` as an empty `ArrayList`.
- A copy constructor. (Make a good decision about what kind of copy you are making.)
- A correctly written `equals` method that overrides the method inherited from the `Object` class. To be considered equal, two `WordSets` must contain the same collection of words, but they do not have to be in the same order.

[This page is intentionally blank.]

27. (8 points) Write the method `linearizer` with the prototype below. The return value will be a one-dimensional array containing references to the same `Strings` that are contained in the parameter. (Combine the elements from all of the rows into an ordinary array). You may not use `ArrayList` or any other Java collection.

```
public static String[] linearizer(String[][] a) {
```