Mr Hammond Wed 17th Sept, 2014

Name: _____

In questions 1 through 9, place one of the following words (or pairs of words) in each blank space, as appropriate: algorithm, bytecode, compiler, embedded, extendible, private, robust, runtime system, semantics, source code, syntax.

1. [5 points] The original promoters of Java imagined it as supporting the development of software to be _____ in small devices, such as a coffee maker.

2. [5 points] The word _____ refers to the *meaning* of Java code, or what it actually does in human, real-world terms.

3. [5 points] Software is called _____ if it has been engineered to "gracefully" handle unexpected inputs or other aspects of the software's environment.

4. [5 points] Instance variables (aka "fields" or "instance fields") should always be marked with the Java keyword _____ .

5. [10 points] The Java _____ is a piece of software that takes _____ as input and produces bytecode as output.

6. [5 points] As a software engineer, you want to construct software that is easily _____ in case it is called upon to perform tasks that were not originally thought of by the client.

7. [5 points] To please the Java compiler, you must use correct Java _____ .

8. [10 points] The Java _____ takes _____ as input to produce the user experience of the software product.

9. [5 points] Software engineers will generally prefer a simple but efficient _____ – a detailed method for solving a problem or performing a task.

In questions 10 through 14, write the declaration – and, possibly, initialization – requested.

10. [5 points] Declare a local variable named `famousLastWords` of type "reference to an object of the `String` class."

---

11. [5 points] Declare an instance variable of type `boolean`, named `hasFeathers`, initialized to the value `true`.

---

12. [5 points] Declare a local variable of type `double`, named `velocity`, initialized to the value `60.0`.

---

13. [5 points] Declare an instance variable of type `int`, named `begYear`, given the initial value `1949`.

---

14. [3 points (bonus)] Declare a local variable named `mammal` of type "reference to an object of the `Mammal` class."

In questions 15 through 17, write the method header requested by filling in the blanks.

15. [8 points] Method named `getVelocity` which returns a `double` and takes no arguments [inputs].

    `public` _____ _____ `()`

16. [8 points] Method named `getName` which returns a reference to an object of the `String` class and takes no arguments.

    `public` _____ _____ `()`

17. [16 points] Method named `setDistance` which returns no value and takes an argument named `newDistance` of type `double`.

    `public` _____ _____ `(` _____ _____ `)`

18. [5 points] In Java, the value of the expression `15 / 6` is _____ .

19. [3 points (bonus)] In Java, the value of the expression `15 % 6` is _____ .

20. Is the proposed name in compliance with *both* Java syntax rules and Java programming convention (style)? Write "yes" or "no".

    (a) [5 points] a class named `aardvark` _____

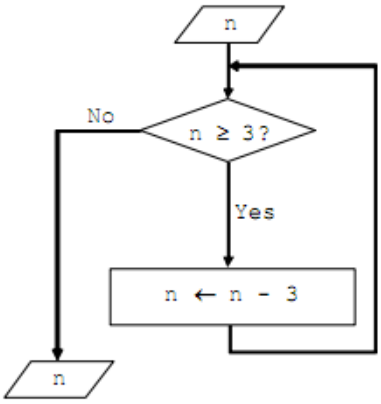
    (b) [5 points] a method named `GetNotes` _____

    (c) [5 points] a variable named `numItems` _____

    (d) [5 points] a variable named `myfavoritetoy` _____

    (e) [5 points] a method named `printNotes` _____

21. [10 points] What is the output from the algorithm shown in the flowchart below when the input is $n = 17$?



    A. 14 B. 5 C. 2 D. -1

22. [5 points] Points for good style!

Name: _____

1. 25 points Ms. Frizzle gives a test. Scores on the test are not higher than 65 points and not lower than 0 points. Grades are assigned as follows – s represents the point score:

$$\begin{aligned} 58.5 \leq s \leq 65 &\Rightarrow \text{A} \\ 52 \leq s < 58.5 &\Rightarrow \text{B} \\ 45.5 \leq s < 52 &\Rightarrow \text{C} \\ 39 \leq s < 45.5 &\Rightarrow \text{D} \\ 0 \leq s < 39 &\Rightarrow \text{F} \end{aligned}$$

The `getGrade` method takes the score `s` as an argument, and returns the grade associated with that score. (The grade is a `String` containing the single appropriate letter.) Write the missing code for the `getGrade` method in the space below, using a single `if ... else if ... else` sequence.

```
public String getGrade(double s)
{
   String g = ""; // grade, to be returned




   return g;
}
```

2. [20 points] Billy helps his neighbors in inclement weather. The amount he makes in a day depends on whether it snows and/or rains on a particular day. His daily earnings are given in the following table:

	snow	no snow
rain	$200	$50
no rain	$100	$2 (tip)

The method `billysDough` returns Billy's earnings, given the weather condition. Write the missing code, using *nested* `if ... else` logic.

```
public int billysDough(boolean snows, boolean rains)
{
   int earn; // Billy's earnings, returned
```

```
   return earn;
}
```

3. [15 points] The method `ranMonth` returns a random integer r such that $1 \leq r \leq 12$. Write the body of `ranMonth` below, using `Math.random()`.

```
public int ranMonth()
{









}
```

4. Variables `s1` and `s2` are declared and initialized as follows:

```
String s1 = "I␣love␣mac␣and␣cheese.",
       s2 = "I␣hate␣fish␣and␣chips.",
       s3 = ... ;
```

You do not know the contents of `s3`, but this `String` is *guaranteed* to contain the word "blech".

(a) Evaluate the following expressions. In some cases you may only be able to say "a positive `int`" or "a negative `int`."

i. [4 points] `s1.length()`

value is _____

ii. [4 points] `s2.length() - s1.length()`

value is _____

iii. [4 points] `s2.length() == s1.length()`

value is _____

iv. [4 points] `s1.substring(17)`

value is _____

v. [4 points] `s1.substring(17,18)`

value is _____

vi. [4 points] `s2.substring(2,5)`

value is _____

vii. [4 points] `s1.compareTo(s2)`

value is _____

viii. [4 points] `s1.equals(s2)`

value is _____

ix. [4 points] `s2.compareTo(s2)`

value is _____

(b) Write code to do the following:

i. [6 points] declare the variable `psn` as an `int`, and initialize it to the position of the word "blech" in `s3`.

ii. [6 points] using your `psn` variable, write an expression which returns the word "blech" (and only that word) from `s3`.

iii. [6 points] write an expression which returns the phrase "mac and chips" extracted from the existing `String` variables.

5. Given the following code:

```
int i = 3, j = 6, k = 10;
double x = -1.7, y = 2.3, z = 0.2;
```

State the Java type and value of each of the following expressions.

(a) [6 points]

```
j / i
```

type is _____, value is _____

(b) [6 points]

```
k % i
```

type is _____, value is _____

(c) [6 points]

```
(int)x
```

type is _____, value is _____

(d) [6 points]

```
(int)z
```

type is _____, value is _____

(e) [6 points]

```
x > i
```

type is _____, value is _____

(f) [6 points]

```
(i + 5) <= k
```

type is _____, value is _____

(g) [6 points]

```
(int)y + 5
```

type is _____, value is _____

6. If `a` and `b` are `boolean` expressions, then – write *yes* or *no* –
   (a) [6 points] if `a` is `false` and `a || b` is evaluated, will `b` be evaluated? _____
   (b) [6 points] if `a` is `false` and `a && b` is evaluated, will `b` be evaluated? _____
7. [10 points] Points for good style!

Mr Hammond Thu 18th Dec, 2014

Name: _____

In questions 1 through 19, place one of the following words (or pairs of words) in each blank space, as appropriate:

abstract class	abstract method	algorithm	bit	byte
bytecode	compiler	constructor	embedded	extendible
HAS-A	inner loop	interface	IS-A	outer loop
private	robust	runtime system	semantics	source code
static method	subclass	superclass	syntax	

1. [10 points] In the code below, the `row` loop is referred to as the _____ and the `col` loop is referred to as the _____ .

```
for (int row = 0; row < arr.length; row++)
{
   for (int col = 0; col < arr[0].length; col++)
   {
      // loop body
   }
}
```

2. [3 points] The Java keyword `new` is used in invoking a class's _____ , a special kind of method which returns a reference to a newly-created object of the class.

3. [10 points] When one class `extend`s another, a(n) _____ relationship is established. When one class *contains* an object of its own class or of another class, a(n) _____ relationship is established.

4. [3 points] The original promoters of Java imagined it as supporting the development of software to be _____ in small devices, such as a coffee maker.

5. [10 points] When a class header includes the code `Mammal extends Animal`, it means that `Mammal` is the _____ and `Animal` is the _____ .

6. [3 points] A "class method" – also referred to as a "_____ " – does not depend on the existence of any object(s) of the class for the method to be called. (An example is `Math.random()`.)

7. [3 points] The word _____ refers to the *meaning* of Java code, or what it actually does in human, real-world terms.

8. [3 points] Software is called _____ if it has been engineered to "gracefully" handle unexpected inputs or other aspects of the software's environment.

9. [3 points] Instance variables (aka "fields" or "instance fields") should by convention always be marked with the Java keyword _____ .

10. [5 points] A(n) _____ is a method with no body, that is, with no definition in code.

11. [6 points] The Java _____ is a piece of software that takes _____ as input and produces bytecode as output.

12. [5 points] The word _____ is shorthand for "binary digit."

13. [5 points] A(n) _____ contains of eight (8) binary digits.

14. [3 points] As a software engineer, you want to construct software that is easily _____ in case it is called upon to perform tasks that were not originally thought of by the client.

15. [5 points] A(n) _____ is a special kind of class which consists of nothing but abstract methods and public constants. This kind of class must be `implement`ed by another class.

16. [3 points] To please the Java compiler, you must use correct Java _____ .

17. [6 points] The Java _____ takes _____ as input to produce the user experience of the software product.

18. [3 points] Software engineers will generally prefer a simple but efficient _____ – a detailed method for solving a problem or performing a task.

19. [10 points] A(n) _____ is a class containing at least one method with no body, that is, with no definition in code.

In questions 20 through 24, write the declaration – and, possibly, initialization – requested.

20. [3 points] Declare a local variable named `messyRoom` of type "reference to an object of the `MessyPlace` class." _____

21. [3 points] Declare an instance variable of type `boolean`, named `hasSkin`, initialized to the value `false`. _____

22. [3 points] Declare a local variable of type `double`, named `position`, initialized to the value `6.0`. _____

23. [3 points] Declare an instance variable of type `int`, named `numMonths`, given the initial value `12`. _____

24. [5 points] Declare a local variable named `yak` of type "reference to an object of the `Mammal` class." _____

In questions 25 through 27, write the method header requested by filling in the blanks.

25. [8 points] Method named `getVelocity` which returns a `double` and takes no arguments [inputs].

`public` _____ _____ `()`

26. [8 points] Method named `getName` which returns a reference to an object of the `String` class and takes no arguments.

`public` _____ _____ `()`

27. [16 points] Method named `setDistance` which returns no value and takes an argument named `newDistance` of type `double`.

`public` _____ _____ `(` _____ _____ `)`

28. Is the proposed name in compliance with *both* Java syntax rules and Java programming convention (style)? Write "yes" or "no".

    (a) [3 points] a class named `Exam` _____

    (b) [3 points] a method named `GetNotes` _____

    (c) [3 points] a variable named `numItems` _____

    (d) [3 points] a variable named `MyFavoriteToy` _____

    (e) [3 points] a method named `printNotes` _____

    (f) [3 points] a constructor named `Circle` _____

29. Complete the code according to the instructions, using `Math.random()`.

    (a) [10 points] The variable `i` takes on a random integer value such that $0 \leq$ `i` ≤ 9.

    `int i =` _____ ;

    (b) [10 points] The variable `j` takes on a random integer value such that $5 \leq$ `j` ≤ 17.

    `int j =` _____ ;

30. Variables `s1` and `s2` are declared and initialized as follows:

```
String s1 = "APCS is easy!",
       s2 = "APCS is not hard!";
```

(a) Evaluate the following expressions. In some cases you may only be able to say "a positive `int`" or "a negative `int`."

i. [3 points] `s1.length()`

value is _____

ii. [3 points] `s2.length() - s1.length()`

value is _____

iii. [3 points] `s2.length() == s1.length()`

value is _____

iv. [3 points] `s2.substring(8)`

value is _____

v. [3 points] `s1.substring(4,6)`

value is _____

vi. [3 points] `s1.substring(5)`

value is _____

vii. [3 points] `s1.compareTo(s2)`

value is _____

viii. [3 points] `s1.equals(s2)`

value is _____

ix. [3 points] `s2.compareTo(s2)`

value is _____

31. Given the following code:

```
int i = 2, j = 7, k = 12;
double x = -2.7, y = 5.3, z = 0.4;
```

State the Java type and value of each of the following expressions.

(a) [4 points]

```
j / i
```

type is _____ , value is _____

(b) [4 points]

```
k % i
```

type is _____ , value is _____

(c) [4 points]

```
(int)x
```

type is _____ , value is _____

(d) [4 points]

```
(int)z
```

type is _____ , value is _____

(e) [4 points]

```
x > i
```

type is _____ , value is _____

(f) [4 points]

```
(i + 5) <= k
```

type is _____ , value is _____

(g) [4 points]

```
(int)y + 5
```

type is _____ , value is _____

32. [24 points] The header for method `mystery` is given below. This method is passed a reference to a two-dimensional array of `int` which follows the convention that the first dimension is the row and the second dimension is the column.

    The method returns the total – *not* the count – of all the elements in `arr` that are in *both* an even-numbered (0,2,4,6,...) row and an odd-numbered (1,3,5,7,...) column. Complete the code for method `mystery`.

    If you need to make notes, do so at the bottom of this page. Your code should be quite neat.

```
// return total of elements in arr which are
// in an even-numbered row and an odd-numbered column
public int mystery(int[][] arr)
{









}
```

33. [10 points] Points for good style!

Name: _____

If you need to make notes, do so outside the space for your code. Your code should be quite neat.

1. [20 points] Write the body of the method described in the comment.

```
/**
 * Given two strings, append them together (known
 * as "concatenation") and return the result. However,
 * if the strings are different lengths, omit chars
 * from the end of the longer string so it is the
 * same length as the shorter string. So "Hello" and
 * "Hi" yield "HeHi". The strings may be any length.
 *
 * minConcat("Hello", "Hi") --> "HeHi"
 * minConcat("Hello", "java") --> "Helljava"
 * minConcat("java", "Hello") --> "javaHell"
 */
public int minConcat(String a, String b)
{









}
```

2. [20 points] Write the body of the method described in the comment.

```
/**
 * Given a string, if the string begins with "green" or
 * "white" return the string "Tommy!", otherwise return
 * the empty string.
 *
 * schSpirit("whitexx") --> "Tommy!"
 * schSpirit("xxgreen") --> ""
 * schSpirit("greenTimes") --> "Tommy!"
 */
public String schSpirit(String str) {









}
```

3. [20 points] Write the body of the method described in the comment.

```
/**
 * Given 2 strings, a and b, return a string of the form
 * long+short+long, with the longer string on the outside
 * and the shorter string on the inside. The strings will
 * not be the same length, but they may be empty (length 0).
 *
 * mamboString("Hello", "hi") --> "HellohiHello"
 * mamboString("hi", "Hello") --> "HellohiHello"
 * mamboString("aaa", "b") --> "aaabaaa"
 */
public String mamboString(String a, String b) {









}
```

4. [20 points] Write the body of the method described in the comment.

```
/**
 * Given a string, if the first or last chars are 'E', return
 * the string without those 'E' chars, otherwise return the
 * string unchanged.
 *
 * withoutE("EHiE") --> "Hi"
 * withoutE("EHi") --> "Hi"
 * withoutE("HxiE") --> "Hxi"
```

```
}
```

5. [10 points] Points for good style!

Name: _____

If you need to make notes, do so outside the space for your code. Your code should be quite neat.

0. [20 points] Write the body of the method described in the comment.

```
/**
 * Given an int array, return true if the array contains
 * 3 twice, or 7 twice. The array will be length 0, 1, or 2.
 *
 * double37({3, 3}) --> true
 * double37({7, 7}) --> true
 * double37({3, 7}) --> false
 **/
public boolean double37(int[] nums) {









}
```

1. [20 points] Write the body of the method described in the comment.

```
/**
 * Start with 2 int arrays, a and b, each length 2. Consider the
 * sum of the values in each array. Return the array which has
 * the smaller sum. In event of a tie, return a.
 *
 * smallerTwo({1, 2}, {3, 4}) --> {1, 2}
 * smallerTwo({3, 4}, {1, 2}) --> {1, 2}
 * smallerTwo({1, 1}, {0, 2}) --> {1, 1}
 */
public int[] smallerTwo(int[] a, int[] b) {









}
```

2. [20 points] Write the body of the method described in the comment.

```
/**
 * Given an array of ints, swap the first and last elements in
 * the array. Return the modified array. The array length will
 * be at least 1.
 *
 * swapEnds({1, 2, 3, 4}) --> {4, 2, 3, 1}
 * swapEnds({1, 2, 3}) --> {3, 2, 1}
 * swapEnds({8, 6, 7, 9, 5}) --> {5, 6, 7, 9, 8}
 */
public int[] swapEnds(int[] nums) {









}
```

3. [20 points] Write the body of the method described in the comment.

```
/**
 * Given an array of ints of odd length, look at the first, last,
 * and middle values in the array and return the smallest. The
 * array length will be a least 1.
 *
 * minTriple({1, 2, 3}) --> 1
 * minTriple({5, 3, 2}) --> 2
 * minTriple({5, 2, 3}) --> 2
 */
public int minTriple(int[] nums) {









}
```

4. [10 points] Points for good style!

Name: _____

If you need to make notes, do so outside the space for your code. Your code should be quite neat.

0. [20 points] Write the body of the method described in the comment.

```
/**
 * When bunnies get together for a party, they like to have
 * squirt guns. A bunny party is successful when the number
 * of squirt guns is between 20 and 40, inclusive. Unless it
 * is the weekend, in which case there is no upper bound on
 * the number of squirt guns. Return true if the party with
 * the given values is successful, or false otherwise.

 * squirtGunParty(10, false) --> false
 * squirtGunParty(30, false) --> true
 * squirtGunParty(50, true) --> true
 */
public boolean squirtGunParty(int sqGuns, boolean isWeekend) {









}
```

1. [20 points] Write the body of the method described in the comment.

```
/**
 * Given a number n, return true if n is in the range 16..25,
 * inclusive. Unless "outsideMode" is true, in which case
 * return true if the number is less or equal to 16, or greater
 * or equal to 25.
 *
 * in16To25(21, false) --> true
 * in16To25(26, false) --> false
 * in16To25(26, true) --> true
 */
public boolean in16To25(int n, boolean outsideMode) {









}
```

2. [20 points] Write the body of the method described in the comment.

```
/**
 * Given two int values, return whichever value is larger.
 * However if the two values have the same remainder when
 * divided by 4, then return the smaller value. However, in
 * all cases, if the two values are the same, return 0.
 *
 * maxMod4(2, 3) --> 3
 * maxMod4(6, 2) --> 2
 * maxMod4(3, 3) --> 0
 */
public int maxMod4(int a, int b) {









}
```

3. [20 points] Write the body of the method described in the comment.

```
/**
 * Return the sum of two 6-sided dice rolls, each in the
 * range 1..6. However, if noDoubles is true, if the two
 * dice show the same value, increment one die to the next
 * value, wrapping around to 1 if its value was 6.
 *
 * withoutDoubles(2, 3, true) --> 5
 * withoutDoubles(3, 3, true) --> 7
 * withoutDoubles(3, 3, false) --> 6
 */
public int withoutDoubles(int die1, int die2, boolean noDoubles) {









}
```

4. [10 points] Points for good style!

Make sure your work is legible!

Name: _____

If you need to make notes, do so outside the space for your code. Your code should be quite neat.

0. (`Card` class) This question concerns the `Card` class of the Elevens lab. Work on this class took place in Activity 1.

   (a) [15 points] constructor

   The `Card` class has instance variables defined as follows:

```
/**
 * String value that holds the suit of the card
 */
private String suit;

/**
 * String value that holds the rank of the card
 */
private String rank;

/**
 * int value that holds the point value.
 */
private int pointValue;
```

   Given the header for the constructor for the `Card` class, write the body of the constructor in the space provided.

```
public Card(String cardRank, String cardSuit,
            int cardPointValue) {
    //initializes a new Card with the given rank,
    //   suit, and point value









}
```

(b) [15 points] `pointvalue()` method

The class has an accessor method for the `Card` object's point value. Given the header, write the body of this method in the space provided.

```
public int pointValue() {




}
```

(c) [15 points] `matches()` method

The class has a method which determines whether the `Card` object matches another `Card` object. "Matches" means that the two `Card` objects have the same rank, suit, and point value. (Remember, you can use the `==` operator for `int`s, but not for `String`s.) Given the header, write the body of this method in the space provided.

```
public boolean matches(Card otherCard) {




}
```

1. (`Deck` class) This question concerns the `Deck` class of the Elevens lab. Work on this class took place in Activity 2.

   (a) [15 points] constructor

   The `Deck` class has instance variables defined as follows:

```
/**
 * cards contains all the cards in the deck.
 */
private List<Card> cards;

/**
 * size is the number of not-yet-dealt cards.
 * Cards are dealt from the top (highest index) down.
 * The next card to be dealt is at size - 1.
 */
private int size;
```

   Given the header for the constructor for the `Deck` class, write the body of the constructor in the space provided. You may remember that the `ranks` and `values` arrays must have the same length, but the `suits` array can be a different length. The deck contains one of each possible rank-suit combination. You need to construct the `ArrayList` to contain the cards, construct and place all the cards in the list, and initialize the `size` instance variable. Just before exiting the constructor, call `shuffle()`.

```
public Deck(String[] ranks, String[] suits, int[] values) {




}
```

(b) [15 points] `size()` method

This method returns the number of undealt cards in the deck. Given the header, write the body in the space provided.

```
public int size() {



}
```

2. [15 points] (`ElevensBoard` class) This question concerns the `containsJQK` method of the `ElevensBoard` class of the Elevens lab. Work on this class took place in Activity 9.

   The `containsJQK` method takes as an argument a list of `Integer` objects which can have any number of elements. It is important to remember that the elements in the list represent board positions. In this implementation of the Elevens game, the "face card" objects have ranks "jack", "queen", and "king". The expression `cardAt(i)` returns the `Card` object occupying board position `i`.

   Given the method header, write the body in the space provided.

```
private boolean containsJQK(List<Integer> selectedCards) {

}
```

3. [10 points] Points for good style!

Make sure your work is legible!

Name: _____

If you need to make notes, do so outside the space for your code. Your code should be quite neat.

0. [15 points] Write the body of the method described in the comment.

```
/**
 * Return the number of odd ints in the given array.
 * Note: the % "mod" operator computes the remainder,
 *    e.g. 5 % 2 is 1.
 *
 * countOdds({3, 2, 3, 4, 5}) --> 3
 * countOdds({3, 3, 1}) --> 3
 * countOdds({2, 4, 6}) --> 0
 */
public int countOdds(int[] nums) {









}
```

1. [15 points] Write the body of the method described in the comment.

```
/**
 * Given an array of ints, return true if the array contains a 5
 * next to a 5 somewhere.
 *
 * has55({1, 5, 5}) --> true
 * has55({1, 5, 1, 5}) --> false
 * has55({5, 1, 5}) --> false
 */
public boolean has55(int[] nums) {









}
```

2. [15 points] Write the body of the method described in the comment.

```
/**
 * Given an array of ints, return true if the number of 0's is
 * greater than the number of 1's, false otherwise.
 *
 * more04({0, 4, 0}) --> true
 * more04({0, 4, 0, 4}) --> false
 * more04({0, 0}) --> true
 */
public boolean more01(int[] nums) {









}
```

3. [15 points] Write the body of the method described in the comment.

```
/**
 * Given an array of scores, return true if each score is
 * equal or greater than the one before. The array will be length
 * 2 or more.
 *
 * scoresIncreasing({1, 3, 4}) --> true
 * scoresIncreasing({1, 3, 2}) --> false
 * scoresIncreasing({1, 1, 4}) --> true
 */
public boolean scoresIncreasing(int[] scores) {









}
```

4. [15 points] Write the body of the method described in the comment.

```
/**
 * Given an array of strings, return the count of the number
 * of strings with length greater than or equal to the given
 * length.
 *
 * wordsCount({"a", "bb", "b", "ccc"}, 2) --> 2
 * wordsCount({"a", "bb", "b", "ccc"}, 3) --> 1
 * wordsCount({"a", "bb", "b", "ccc"}, 4) --> 0
 */
public int wordsCount(String[] words, int len) {









}
```

5. [15 points] Write the body of the method described in the comment.

```
/**
 * Given an array of strings, return a new array containing
 * the first N strings. N will be in the range 1..length.
 *
 * wordsFront({"a", "b", "c", "d"}, 1) --> {"a"}
 * wordsFront({"a", "b", "c", "d"}, 2) --> {"a", "b"}
 * wordsFront({"a", "b", "c", "d"}, 3) --> {"a", "b", "c"}
 */
public String[] wordsFront(String[] words, int n) {









}
```

6. [10 points] Points for good style!