

**COM S 227: Object-Oriented Programming**  
**Exam 2 Practice**  
**Spring 2024**

**DO NOT OPEN THIS EXAM UNTIL INSTRUCTED TO DO SO**

**Student Name:** \_\_\_\_\_

**ISU NetID (username):** \_\_\_\_\_@iastate.edu

**Instructions:**

Closed book/notes, no electronic devices, no headphones. Time limit 75 minutes. Partial credit may be given for partially correct solutions. *If you have questions, please ask!*

- Use correct Java syntax for writing code. You do not need `import` statements.
- You are not required to write comments for your code; however, brief comments may help make your intention clear in case your code is incorrect.
- Use the abbreviation “SOP” for `System.out.println`.
- It is ok to use literal values; you don’t need to define constants for numbers.
- There are some excerpts from the Java API documentation on the last page.
- If you remove any of the last pages, you must turn them in with your exam.

Question	Points
1	/26
2	/34
3	/34
4	/34
5	/42
6	/30
<b>Total</b>	/200

**1. (26 pts)** Write a static method that, given a string, returns a new string with all vowels ("a", "e", "i", "o", "u") replaced with "\_". Only consider lower case letters.

For example, given the string "volume", return the string "v\_l\_m\_".

```
public static String blankVowels(String str) {
```

```
}
```

**2. (34 pts)** The goal of this problem is to implement a method, that given an array of integers, returns a new array containing the same numbers, in the same order, but without duplicates. For example, given the array [5, 4, 5, 6, 4, 2], the method returns [5, 4, 6, 2]. To receive credit, you must implement the method using the strategy that is specified below in the provided spaces.

```
public static int[] removeDuplicates(int[] dupsArr) {
```

a) Declare and initialize an ArrayList that can hold Integer elements using the variable name **noDupsList**.

b) Implement an algorithm to fill **noDupsList** with the values from **dupsArr**, in the same order, but with duplicates removed. For example, if the array contains [5, 4, 5, 6, 4, 2], **noDupsList** should contain [5, 4, 6, 2].

c) Declare and initialize an array called **noDupsArr** that can hold int elements and is the same size as **noDupsList**.

d) Copy each element from **noDupsList** to **noDupsArr**, in the same order.

```
    return noDupsArr;
```

```
}
```

**3. (34 pts)** Suppose you are **given** a class **Grammar** containing a method

```
public static boolean isPalindrome(String s)
```

that returns true if the given string is a palindrome (same forwards and backwards). The goal of this problem is to implement a method, that given the name of a text file, reads the file and returns an ArrayList containing all words in the file that are palindromes.

The format of the file is words separated by single spaces. For example, the file containing: "we will kayak at noon madam" should return the list ["kayak", "noon", "madam"]. To receive credit, you must implement the method using the strategy that is specified below in the provided spaces.

```
public static ArrayList<String> readPalidromes(String fileName) {
```

a) Declare and initialize an ArrayList that can hold String elements using the variable name **palindromeList**.

b) Open the file for reading by creating a Scanner object. The exception handling has been done for you, complete the following code where appropriate.

```
try {
```

```
    } catch (FileNotFoundException e) {  
        // not able to open file, return an empty list  
        return palindromeList;  
    }
```

c) Read the words from the file and put the palindromes into the **palindromeList**. Do not write the **isPalindrome** method, just assume the class **Grammar** is available for you to use. **You will not receive full points** for not using **isPalindrome** or implementing your own version of **isPalindrome**. Don't worry about imports.

```
    return palindromeList;
```

```
}
```

**4. (34 pts)** The goal of this problem is to implement a method that, given a 1D int array `arr` of size  $n$ , returns a new 2D array of size  $n \times n$  in which the  $r$ -th row of `arr` is shifted to the right  $r$  positions. Shifting right by  $r$  positions means each element is moved to its index plus  $r$ , while elements that go beyond the bounds of the array are ignored and positions with no element moved into them are filled with a 0.

For example, given the array:

1	2	4	8
---	---	---	---

The method returns:

1	2	4	8
0	1	2	4
0	0	1	2
0	0	0	1

To receive credit, you must implement the method using the strategy that is specified below in the provided spaces.

```
public static int[][] rightShifts(int[] arr) {
```

a) Declare and initialize a 2D array of int with appropriate dimensions with the name `shiftTable`.

b) Fill `shiftTable` with values as specified in the above problem description.

```
    return shiftTable;
```

```
}
```

**5. (42 pts)** Suppose that a text file contains lines with a name and phone number having the format:

name, xxx-xxx-xxxx

Here is an example file:

John, 123-234-3456

Kim, 122-222-2222

Assume you are given a class called `Contact` with the following constructor and methods.

```
Contact(String givenName, String givenPhoneNumber) // creates a new Contact
String getName() // returns the name as a String
String getPhoneNumber() // returns phone number as a String
```

Create a class `ContactDirectory` suitable for storing a list of `Contact` objects read from a file of the specified file format. To receive credit, you must implement the public methods specified below in the provided spaces. You may not add any extra helper methods. *Do not provide the implementation for `Contact`, assume it is available to use, only show `ContactDirectory`.*

```
public class ContactDirectory{
```

a) Declare and initialize any instance variables if needed.

```
    public ContactDirectory() {

    }

}
```

b) Implement a method to add a given **Contact** object to the directory.

```
    void addContact(Contact c) {
```

```
}
```

c) Implement a method to read a file of the above format, given by its file name. Store each new contact in the directory.

```
void addFromFile(String filename) throws FileNotFoundException {
```

```
}
```

d) Implement a method to get the phone number for a given name. Return an empty string "" if the name is not found.

```
String lookUpPhoneNumber(String name) {
```

```
}
```

6. (30 pts) a) Given the recursive method `confound` below, show all output printed for the method call `confound(5)`. Write down **just the output** as it would be printed.

```
public static int confound(int x) {  
    if (x <= 2) {  
        System.out.println("peach");  
        return x;  
    }  
  
    int y = x - 1;  
    int result = confound(y) + y;  
    System.out.println(result);  
    return result;  
}
```

Output when `confound(5)` is called.

b) Given the recursive method `check` below, show all output printed for the method call `check("aaba")`. Write down **just the output** as it would be printed.

```
public static boolean check(String str) {  
    if (str.length() == 0) {  
        return true;  
    }  
  
    boolean result = (str.charAt(0) == 'a' && check(str.substring(1, str.length())))  
        || (str.charAt(str.length() - 1) == 'b' && check(str.substring(0, str.length() - 1)));  
  
    System.out.println("For string " + str + " result is " + result + ".");  
    return result;  
}
```

Output when `check("aaba")` is called.



Page intentionally left blank for additional space.

## Excerpts from the Java API documentation

### Excerpt from documentation for `java.util.ArrayList<E>`

Method Summary	
<code>boolean add(E element)</code>	Appends the specified element to the end of this list.
<code>boolean add(int index, E element)</code>	Appends the specified element at the specified position in this list.
<code>void clear()</code>	Removes all the elements from this list.
<code>boolean contains(E element)</code>	Returns true if this list contains the specified element.
<code>E get(int index)</code>	Returns the element at the specified position in this list.
<code>E remove(int index)</code>	Returns the element at the specified position in this list (elements to right shift down).
<code>boolean remove(Object obj)</code>	Removes the first occurrence of the specified object, returning false if it does not occur.
<code>E set(int index, E element)</code>	Replaces the element at the given position in this list with the specified element.
<code>int size()</code>	Returns the number of elements in this list.

### Excerpt from documentation for `java.lang.String`

Method Summary	
<code>char charAt(int index)</code>	Returns the character at the given index.
<code>boolean contains(String s)</code>	Returns true if the given string is a substring of this string.
<code>boolean equals(String other)</code>	Returns true if this string is the same as the given string
<code>int indexOf(char ch)</code>	Returns the index of the first occurrence of the given character in this string. Returns -1 if the this string does not contain the given character.
<code>int indexOf(String s)</code>	Returns the index of the first occurrence of the given string s as a substring of this string. Returns -1 if s is not a substring.
<code>int lastIndexOf(char ch)</code>	Returns the index of the last occurrence of the given character in this string. Returns -1 if the this string does not contain the given character.
<code>int lastIndexOf(String s)</code>	Returns the index of the last occurrence of the given string s as a substring of this string. Returns -1 if s is not a substring.
<code>int length()</code>	Returns the length of this string.
<code>String substring(int beginIndex)</code>	Returns a substring of this string starting at beginIndex.
<code>String substring(int beginIndex, int endIndex)</code>	Returns a substring of this string consisting of the characters from beginIndex through endIndex - 1.
<code>String toUpperCase()</code>	Returns a copy of this string with all alphabetic characters converted to upper case.
<code>String toLowerCase()</code>	Returns a copy of this string with all alphabetic characters converted to lower case.
<code>String trim()</code>	Returns a copy of this string with leading and trailing whitespace removed.

**Excerpt from documentation for java.lang.Math**

<b>Method Summary</b>	
<code>static double abs(double x)</code>	Returns the absolute value of the given number.
<code>static int abs(int x)</code>	Returns the absolute value of the given integer.
<code>static double max(double x, double y)</code>	Returns the larger of the two given numbers.
<code>static int max(int x, int y)</code>	Returns the larger of the two given integers
<code>static double min(double x, double y)</code>	Returns the smaller of the two given numbers.
<code>static int min(int x, int y)</code>	Returns the smaller of the two given integers
<code>static double pow(double x, double y)</code>	Returns x to the power y.
<code>static long round(double x)</code>	Returns the given number rounded to the nearest whole number value (can be cast to int)
<code>static double sqrt(double x)</code>	Returns the square root of the given number.

**Excerpt from documentation for java.lang.Integer**

<b>Method Summary</b>	
<code>static int parseInt(String s)</code>	Converts the given string to an int value, if possible.

**Excerpt from documentation for java.lang.Double**

<b>Method Summary</b>	
<code>static double parseDouble(String s)</code>	Converts the given string to a double value, if possible.

**Excerpt from documentation for java.util.Random**

<b>Constructor Summary</b>	
<code>Random()</code>	Constructs a new random number generator.
<b>Method Summary</b>	
<code>int nextInt(int max)</code>	Returns a pseudorandom, uniformly distributed value between 0 (inclusive) and <b>max</b> (exclusive), drawn from this generator's sequence.

**Excerpt from documentation for java.io.File**

<b>Constructor Summary</b>	
<code>File(String filename)</code>	Constructs a new file instance.
<b>Method Summary</b>	
<code>String getName()</code>	Returns the name of the file or directory represented by this object.
<code>boolean isDirectory()</code>	Returns true if this object represents a directory.
<code>File[] listFiles()</code>	If this object represents a directory, returns a list of File objects representing its immediate contents; otherwise returns null

**Excerpt from documentation for java.util.Scanner**

<b>Constructor Summary</b>	
<code>Scanner(InputStream source)</code>	Constructs a new scanner that provides values scanned from a given input stream, such as <code>System.in</code> .
<code>Scanner(String source)</code>	Constructs a new scanner that uses the given string as its input stream.
<b>Method Summary</b>	
<code>boolean hasNext()</code>	Returns true if this scanner has another token in its input.
<code>boolean hasNextDouble()</code>	Returns true if the next token in this scanner's input can be parsed as a floating-point value.
<code>boolean hasNextInt()</code>	Returns true if the next token in this scanner's input can be parsed as an int value.
<code>boolean hasNextLine()</code>	Returns true if this scanner's input contains a newline character.
<code>String next()</code>	Returns the next token in this scanner's input.
<code>double nextDouble()</code>	Returns the next token in this scanner's input, converted to a floating-point value, if possible.
<code>int nextInt()</code>	Returns the next token in this scanner's input, converted to an int value, if possible.
<code>String nextLine()</code>	Returns all input up to the next newline character.
<code>void useDelimiter(String pattern)</code>	Sets this scanner's delimiter to the given string (default delimiter is whitespace).