Beijing National Day School
Department of Mathematics

AP Computer Science A

Test 2: `Arrays` and `ArrayLists`

English Name: ____________________

Pinyin Name: ____________________

Mr. Alwin Tareen, Fall 2018

Exam Record

Part1 ______________/ 20 pts

Part2 ______________/ 15 pts

Part3 ______________/ 12 pts

Total: ______________/ 47 pts

Grade: ______________

AP Computer Science A Test 2: `Arrays` and `ArrayLists` Mr. Alwin Tareen
Fall 2018 BNDS

Part I: Multiple Choice (20 points)

- Determine the answer to each of the following questions, using the available space for any necessary scratchwork.
- Decide which is the best of the choices given, and select the correct answer by placing an "**X**" in the corresponding box.

(1pt) **1.** Which of the following choices correctly declares and initializes an array of integers named `nums`?

□ `integer[] nums = new integer[5];`
□ `new int nums[] = int[5];`
□ `int[] nums = new int[5];`
□ `nums = [5];`

1 pt

(1pt) **2.** What would be the output of the following `Java` code?

```
int[] nums = {2, 4, 6, 8};
System.out.println(nums[0] + " " + nums[1]);
```

□ `2 4`
□ `2 6`
□ `8`
□ `6 8`

1 pt

(1pt) **3.** What would be the output of the following `Java` code?

```
int[] nums = {2, 4, 6, 8};
nums[0] = 23;
nums[3] = nums[1];
System.out.println(nums[0] + " " + nums[3]);
```

□ `31`
□ `23 2`
□ `2 8`
□ `23 4`

1 pt

(1pt) **4.** What is the technical name for the following array declaration in `Java`?

```
double[] scores = {93.7, 86.2, 91.5, 98.3};
```

□ A customizer list.
□ An initializer list.
□ An elementary list.
□ A verifier list.

1 pt

(1pt) **5.** What would be the output of the following `Java` code?

```
double[] nums = { {1.2, 9.0, 3.2},
                  {9.2, 0.5, 1.5},
                  {7.3, 7.9, 4.8} };
System.out.println(nums[2][1]);
```

□ `7.3`
□ `7.9`
□ `9.2`
□ `5.1`

1 pt

5 pts

(1pt) **6.** What would be the output of the following Java code?

```
int[] nums = {12, 34, 9, 0, -62, 88};
System.out.println(nums.length);
```

□ 5
□ 1
□ 6
□ 12

1 pt

(1pt) **7.** Consider the following Java program. We want to assign numbers into the array nums, such that each number is twice the value of its index. Choose one of the following statements to replace `// YOUR CODE HERE` which would generate this result.

```
int[] nums = new int[10];
for (int i = 0; i < nums.length; i++)
{
    // YOUR CODE HERE
}
```

□ `i = 2*i;`
□ `nums[2*i] = 2*i;`
□ `nums[i] = 2*nums[i];`
□ `nums[i] = 2*i;`

1 pt

(1pt) **8.** What would be the output of the following Java code?

```
int[] nums = {2, 4, 6, 8, 10, 1, 3, 5, 7, 9};
for (int i = 0; i < 5; i++)
{
    System.out.print(nums[i] + " ");
}
```

□ 2 4 6 8 10
□ 2 4 6 8
□ 2 4 6 8 10 1
□ 2 4 6 8 10 1 3 5 7 9

1 pt

(1pt) **9.** What would be the output of the following Java code?

```
double[] nums = { {1.2, 9.0},
                  {9.2, 0.5},
                  {7.3, 7.9} };
System.out.println(nums.length);
```

□ 2
□ 4
□ 3
□ 9

1 pt

(1pt) **10.** In Java, elements of an array are automatically initialized to some default value. What is the default value for the elements of an array of integers?

□ 0
□ null
□ true
□ infinity

1 pt

5 pts

(1^{pt}) **11.** Which of the following `import` statements must you provide, if you want to use an `ArrayList` in your `Java` program?

1 pt

□ `import java.util.*`
□ `import collection.util.*`
□ `import Array.List.*`
□ `import java.lang.*`

(1^{pt}) **12.** Which of the following choices is a TRUE statement about an `ArrayList`?

1 pt

□ An `ArrayList` has a fixed size which cannot be altered.
□ An `ArrayList` can only contain primitive data types, not objects.
□ An `ArrayList` is not a data structure.
□ An `ArrayList` can grow or shrink in size.

(1^{pt}) **13.** Consider the following `ArrayList` declaration:

```
ArrayList<String> words = new ArrayList<String>();
```

Which of the following choices would correctly place `"sunshine"` into this `ArrayList`?

1 pt

□ `words.place("sunshine");`
□ `words[0] = "sunshine";`
□ `words.add("sunshine");`
□ `words.container = "sunshine";`

(1^{pt}) **14.** Consider the following `Java` program:

```
ArrayList<String> names = new ArrayList<String>();
names.add("alice");
names.add("bob");
names.add("carl");
```

Which of the following choices would correctly retrieve the name `"carl"` from this `ArrayList`?

1 pt

□ `String student = names.get(2);`
□ `String student = names.get("carl");`
□ `String student = names.retrieve(2);`
□ `Text person = names[2];`

(1^{pt}) **15.** Consider the following `Java` program:

```
ArrayList<String> sports = new ArrayList<String>();
sports.add("basketball");
sports.add("badminton");
sports.add("football");
```

Which of the following choices would correctly delete the sport `"badminton"` from this `ArrayList`?

1 pt

□ `sports.remove(1);`
□ `sports.delete(1);`
□ `sports[1] = null;`
□ `eliminate(sports."badminton");`

5 pts

(1pt) **16.** Consider the following Java program:

```
ArrayList<String> drinks = new ArrayList<String>();
drinks.add("coffee");
drinks.add("tea");
```

Which of the following choices would correctly determine the number of items in this ArrayList?

□ `int num = drinks.quantity();`

□ `int amount = drinks.length;`

□ `int amount = measureItems(drinks);`

□ `int num = drinks.size();`

□ 1 pt

(1pt) **17.** Which of the following choices is a TRUE statement about an ArrayList?

□ You can insert an element anywhere in an ArrayList.

□ You can only add a new element to the beginning of an ArrayList.

□ You can only delete elements from an ArrayList, insertions are not permitted.

□ Whenever you delete an element from an ArrayList, you must add another element to the ArrayList, to maintain balance.

□ 1 pt

(1pt) **18.** Consider the following Java program:

```
ArrayList<String> dinner = new ArrayList<String>();
dinner.add("pizza");
dinner.add("hamburger");
dinner.add("cake");
```

Which of the following choices would correctly replace "hamburger" with "salad"?

□ `dinner.replace(1, "salad");`

□ `dinner.exchange("hamburger", "salad");`

□ `dinner.set(1, "salad");`

□ `dinner["hamburger"] = "salad";`

□ 1 pt

(1pt) **19.** Which of the following choices is the correct way to declare an ArrayList which is intended to hold integers?

□ `ArrayList<Integer> nums = new ArrayList<Integer>();`

□ `ArrayList<int> nums = new ArrayList<int>();`

□ `ArrayList<Amount> nums = new ArrayList<Amount>();`

□ `ArrayList<Quantity> nums = new ArrayList<Quantity>();`

□ 1 pt

(1pt) **20.** Which of the following choices would correctly declare an ArrayList to contain the String data type?

□ `ArrayList<String> words = new ArrayList<String>();`

□ `String[] words = new String[10];`

□ `StringList<words> = new StringList();`

□ `Text<ArrayList> words = new Text<ArrayList>();`

□ 1 pt

□ 5 pts

Part II: Short Answer (15 points)

- Solve each of the following short answer questions. Write your solution in the space provided.

(1^{pt}) **1.** Write a single line of code that will create an array of type `double` called `nums` having 800 elements.

1 pt

(1^{pt}) **2.** Consider the following array: `double[] scores = new double[21]`
What is the output of: `System.out.println(scores.length);`

1 pt

(1^{pt}) **3.** Consider the following array: `int[] temps = {34, 56, -102, 18, 5}`
What is the output of: `System.out.println(temps[1])`?

1 pt

(1^{pt}) **4.** Consider the following array: `int[] temps = {34, 56, -102, 18, 5}`
What is the output of: `System.out.println(temps[3] + temps[4])`?

1 pt

(1^{pt}) **5.** Consider the following array: `String[] names = {"Bob", "Jim", "Sally"}`
Write a single line of Java code that will display the element `"Sally"` from this array.

1 pt

(1^{pt}) **6.** Consider the following array: `int[] nums = new int[7];`
Write a section of Java code that would place the number `23` into every position in the array. *Hint:* Use a `for` loop.

1 pt

(1^{pt}) **7.** Consider the following two-dimensional array: `int[][] nums = new int[10][10];`
Write a section of Java code that would place the number `58` into every position in the array. *Hint:* Use two `for` loops.

1 pt

7 pts

(1^{pt}) **8.** Write a single line of Java code that will instantiate an `ArrayList` object called `sports` and have the restriction that only `String` objects can be stored in it.

1 pt

(1^{pt}) **9.** Consider an `ArrayList` called `holiday` that contains the following `Strings`: `["Happy", "birthday"]`. Write a single line of Java code that will change the contents of this `ArrayList` to: `["Happy", "halloween"]`

1 pt

(1^{pt}) **10.** Consider the following `ArrayList` called `pizza` which contains the following `Strings`: `["pepperoni", "supreme"]`.
What is the output of: `System.out.println(pizza.isEmpty())`?

1 pt

(1^{pt}) **11.** What is the name of the wrapper class which enables a programmer to place `integer` values into an `ArrayList`?

1 pt

(1^{pt}) **12.** What is the name of the wrapper class which enables a programmer to place `double` values into an `ArrayList`?

1 pt

(1^{pt}) **13.** Consider the following `ArrayList` called `drinks` which contains the following `Strings`: `["tea", "coffee"]`.
What is the output of: `System.out.println(drinks.indexOf("juice"))`?

1 pt

(1^{pt}) **14.** Consider the following `ArrayList` called `average` which contains the following `Doubles`: `[98.3, 42.7, 65.4, 35.3, 45.6]`.
What is the output of: `System.out.println(average.size())`?

1 pt

(1^{pt}) **15.** Consider the following `ArrayList` called `amounts` which contains the following `Integers`: `[39, 48, 54, 32, 98, 57]`.
Write a section of Java code that would add up these `Integers` and place the result in the variable: `sum`

1 pt

8 pts

Part III: Java Programming (12 points)

- Show all of your work. Remember that program segments are to be written in the `Java` programming language.

(12^{pts}) **1.** Write a Java class that will simulate a Simple Message System(SMS) text message inbox, similar to that on a typical cellphone.

12 pts

- The class will be called `TextMessage`, and it will consist of an `ArrayList` that will hold multiple SMS text messages. Each SMS text message will be in the form of a `String`.
- The `TextMessage` class should implement the following methods:
  - `addNewArrival`: Adds the incoming SMS text message to the end of the inbox.
  - `messageCount`: Returns the number of SMS text messages in the inbox.
  - `getMessage`: Retrieves the SMS text message at position `i`.
  - `delete`: Deletes the SMS text message at position `i`.
  - `clear`: Clears all SMS text messages from the inbox.
- Implement the `TextMessage` class, with all the necessary attributes and methods. Do not include any other attribute or method apart from the ones previously described.
- The code framework for the `TextMessage` class has been provided for you. Your task is to fill in the necessary code for the instance variables, constructors and methods, to make this a working program.
- *Hint:* Use the supplied `Javadoc` comments above each of the methods to help you write this class.
- The test bench used for the `TextMessage` class has been included below, as well as its respective output to the terminal display.

The Java source code is on the next page.

12 pts

Java Source Code for `TextMessageTest.java`, the Test Bench

```java
import java.util.*;

public class TextMessageTest
{
    public static void main(String[] args)
    {
        ArrayList<String> texts = new ArrayList<String>();
        texts.add("meet me at noon");
        texts.add("we should eat lunch");
        texts.add("i like pizza");

        TextMessage tm = new TextMessage(texts);
        System.out.println(tm);
        tm.addNewArrival("yummy ice cream");
        System.out.println(tm);
        System.out.println("# of texts: " + tm.messageCount() + "\n");
        System.out.println("Text at index 1: " + tm.getMessage(1)+"\n");
        tm.delete(2);
        System.out.println(tm);
        tm.clear();
        System.out.println("# of texts: " + tm.messageCount());
    }
}
```

The Terminal Display Output of `TextMessageTest.java`

```
meet me at noon
we should eat lunch
i like pizza

meet me at noon
we should eat lunch
i like pizza
yummy ice cream

# of texts: 4

Text at index 1: we should eat lunch

meet me at noon
we should eat lunch
yummy ice cream

# of texts: 0
```

Java Source Code for `TextMessage.java`

```
import java.util.*;

/**
 * Representation of an SMS text message inbox.
 * @version 1.0
 */
public class TextMessage
{
    /**
     * Instance variable for the SMS text message inbox.
     */
    private ArrayList<String> inbox;

    /**
     * A constructor that creates a new TextMessage object which
     * represents an SMS text message inbox.
     */
    public TextMessage(ArrayList<String> messages)
    {
        // YOUR CODE HERE





    }

    /**
     * Adds the incoming SMS text message to the end of the inbox.
     *
     * @param textOfSMS  the content of the SMS text message
     */
    public void addNewArrival(String textOfSMS)
    {
        // YOUR CODE HERE





    }

    /**
     * @return the number of SMS text messages in the inbox
     */
    public int messageCount()
    {
        // YOUR CODE HERE





    }
```

0 pts

```
    /**
     * Retrieves the SMS text message at position i.
     *
     * @param i  the index of the SMS text message to be retrieved
     * @return the SMS text message at position i
     */
    public String getMessage(int i)
    {
        // YOUR CODE HERE




    }

    /**
     * Deletes the SMS text message at position i.
     *
     * @param i  the index of the SMS text message to be deleted
     */
    public void delete(int i)
    {
        // YOUR CODE HERE




    }

    /**
     * Clears all SMS text messages from the inbox.
     */
    public void clear()
    {
        // YOUR CODE HERE




    }

    /**
     * Displays all SMS text messages to the output terminal.
     */
    public String toString()
    {
        String result = "";
        for (String item : inbox)
        {
            result += item + "\n";
        }
        return result;
    }
}
```

0 pts

This page is left intentionally blank.

0 pts