

Name: _____ Andrew Id: _____

15-121 Fall 2024 Assessment 1

Up to 50 minutes. No calculators, no notes, no books, and no computers. Show your work!

1. Short Answer.

Answer each of the following in 1 sentence; longer answers will be marked incorrect.

(a) (2 points) What is the difference between a `float` and a `double`?

(b) (1 point) Describe the relationship between a class and an object.

(c) (2 points) Lolwa writes the following function, but finds that it doesn't always work as expected. What is the bug?

```
1  /* Find target in the array. Return true if it is found and false otherwise. */
2  public static boolean stringContains(String[] arr, String target) {
3      for (String s : arr) {
4          if (s == target) {
5              return true;
6          }
7      }
8      return false;
9  }
```

2. (4 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box under the code.

```
public class IncrementorExercise {
    public static void main(String[] args) {
        int a = 10;
        int b = 8;
        int c = 7;

        System.out.println(++b - --a + a-- - ++a - b++ + --a + ++c - c++);
        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
    }
}
```

3. (5 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box under the code.

```
public class CT {
    public String a;
    public int b;
    public static double c;

    public CT(String a, int b) {
        this.a = a;
        this.b = b;
        c = b;
        c = c / a.length();
    }

    public int ds() {
        int ret = 0;
        for (int i = 0; i < this.a.length(); i++) {
            char c = this.a.charAt(i);
            if (c >= '0' && c <= '9') {
                ret++;
            }
        }
        return ret;
    }

    public void mystery(int d, int e) {
        this.b += d;
        this.a = e + this.a;
    }

    public static void main(String[] args) {
        CT f = new CT("Hi", 5);
        CT s = new CT("Heya", 6);
        System.out.println("f: " + f.a + " " + f.b);
        System.out.println("s: " + s.a + " " + s.b);

        f.mystery(10, 123);
        s.mystery(7, 4567);
        System.out.println("f:" + f.ds() + " s:" + s.ds());

        f.c += 1.0;
        s.c += 2.0;

        System.out.println("f: " + f.a + " " + f.b + " " + f.c);
        System.out.println("s: " + s.a + " " + s.b + " " + s.c);
    }
}
```

4. (10 points) **Free Response:**

Write a `Triangle` class with the following properties:

- A triangle can be constructed in one of two ways:
 1. By specifying the length of the three sides, each as a double. For example, calling...
`Triangle t1 = new Triangle(6.0, 4.1, 3.0);`
would create a triangle with sides of length 6, 4.1, and 3.
 2. By specifying the length of two sides. In this case, it is assumed that they are two sides of a right triangle, and the hypotenuse will be calculated. For example, calling...
`Triangle t2 = new Triangle(3.0, 4.0);`
would create a triangle with sides of length 3, 4, and 5. (The 5 was calculated.)
- Triangles are immutable: Once they are created, they never change. (So all instance variables should be private and you do not need to create setters for them.)
- A triangle has a method that can be used to calculate its area.
- When printed, a triangle should print out its dimensions. For example, `System.out.println(t1)` should display: 6.0, 4.1, 3.0. `System.out.println(t2)` should display: 3.0, 4.0, 5.0.
- If invalid sides are given when constructing a triangle, then an `IllegalArgumentException` is thrown. One way that sides could be invalid is if one or more sides is less than or equal to 0. Another way is if the three sides don't satisfy the Triangle Inequality Theorem. For example,
`Triangle t3 = new Triangle(1.0, 2.0, 4.0);` triggers an `IllegalArgumentException`.

If you need a refresher on basic triangle math, please see the last page of this exam.

Extra space for Question 4

5. Free Response

In this problem you will write a variety of methods in a `StringList` class. A `StringList` is used to store and operate on a list of strings.

Consider the following code for the `StringList` class:

```
public class StringList {
    // This class only has two instance variables. You may not add any more.
    private String[] elems;
    private int howm;

    /**
     * Construct a new, empty, String with the specified initial capacity. Initial
     * capacity means the number of items that can be added before a resize must
     * occur.
     *
     * @param size The initial capacity of the StringList.
     */
    public StringList(int size) {
        // You will write this code.
    }

    /**
     * Add a new String to the list, expanding the underlying array if needed. If
     * the array does need to be expanded, then double its size prior to adding the
     * new element.
     *
     * @param s The new String to add to the list. It should not be null. If it is,
     *           then add nothing to the list.
     */
    public void add(String s) {
        // You will write this code.
    }

    /**
     * Find the longest string in the list and returns its index in the list.
     *
     * For example, if the list contains ["Hi", "Ahmed is great", "Philly"] then
     * this method would return 1.
     *
     * @return The index of the longest string. Return -1 if the list is empty.
     */
    public int findLongest() {
        // You will write this code.
    }

    /**
     * Find and remove the longest string from the list. This does not return
     * anything.
     */
    public void removeLongest() {
        // You will write this code.
    }
}
```

(a) (2 points) Write the constructor, as specified below.

```
/**
 * Construct a new, empty, String with the specified initial capacity. Initial
 * capacity means the number of items that can be added before a resize must
 * occur.
 *
 * @param size The initial capacity of the StringList.
 */
public StringList(int size) {
```

(b) (5 points) Write the method `add`, as specified below.

```
/**
 * Add a new String to the list, expanding the underlying array if needed. If
 * the array does need to be expanded, then double its size prior to adding the
 * new element.
 *
 * @param s The new String to add to the list. It should not be null. If it is,
 *           then add nothing to the list.
 */
public void add(String s) {
```

(c) (5 points) Write the method `findLongest`, as specified below.

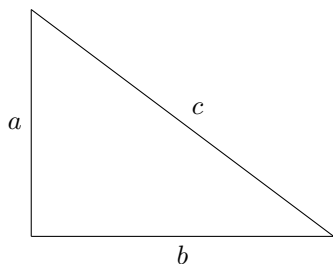
```
/**
 * Find the longest string in the list and returns its index in the list.
 *
 * For example, if the list contains ["Hi", "Ahmed is great", "Philly"] then
 * this method would return 1.
 *
 * @return The index of the longest string. Return -1 if the list is empty.
 */
public int findLongest() {
```

(d) (4 points) Write the method `removeLongest`, as specified below.

```
/**
 * Find and remove the longest string from the list. This does not return
 * anything.
 */
public void removeLongest() {
```

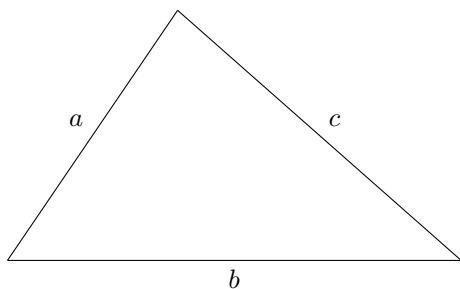
Triangle Refresher

- When given two sides of a right triangle, the hypotenuse can be computed using the Pythagorean Theorem. For any valid right triangle with sides of length a , b , and c ...



$$c = \sqrt{a^2 + b^2}$$

- The Triangle Inequality Theorem states that the sum of any two sides of a triangle should always be larger than the third side. For any valid triangle with sides of length a , b , and c ...

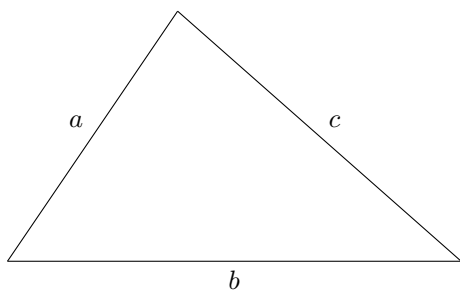


$$a + b > c$$

$$a + c > b$$

$$b + c > a$$

- Heron's formula allows you to calculate the area of a triangle based only on the lengths of its sides. For any valid triangle with sides of length a , b , and c ...



$$\text{Area} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{where } s = \frac{a+b+c}{2}$$