UNIVERSITY OF WISCONSIN-LA CROSSE
Department of Computer Science

CS 120 **Software Design I** **Spring 2017**
Midterm Exam 01 **28 February 2017**

- **Do not turn the page until instructed to do so.**
- **This booklet contains 10 pages including the cover page.**
- **This is a closed-book exam. All you need is the exam and a writing utensil. (You may use a calculator if you wish.)**
- **You have exactly 55 minutes.**
- **The maximum possible score is 50.**

PROBLEM	SCORE
1	
2	
3	
4	
5	
6	
TOTAL	

NAME: _____

1. **(8 pts.) TRUE/FALSE.**

   For each of the following, indicate whether the statement is true or false.
   You do not need to explain your answers.

   a. If the left-hand target of an assignment statement is an `int` variable, then the expression on the right-hand side will always be evaluated using integer arithmetic.

   b. The following two lines of code do the same thing:

```
String x = 1 + 2 + "3";
String x = "1" + 2 + 3;
```

   c. The following two lines of code do the same thing:

```
String x = 1 + "2" + 3;
String x = "1" + 2 + 3;
```

   d. A Java class diagram shows us the names of methods, along with their inputs and return type; this is all the information we need to know what those methods do, exactly.

   e. When doing arithmetic in Java, multiplication (`*`) has higher precedence than division (`/`).

   f. The following line of code will not compile, due to an error:

```
int x = 'z' + 3;
```

   g. The following line of code will not compile, due to an error:

```
char x = 'z' + 3;
```

   h. The following code will always print some result, *no matter what* value the integer `x` has:

```
if( x >= 1 )
    System.out.println( "One" );
else if ( x == 0 )
    System.out.println( "Two" );
```

2. **(8 pts.) SHORT ANSWER.**

Circle the appropriate answer and fill in the blanks where required.

a. (*4 pts.*) An assignment like the following:

```
int x = (int) 3 / 2.0;
```

(WILL / WON'T) compile properly, because _____

_____.

An assignment like the following:

```
int x = (int)( 3 / 2.0 );
```

(WILL / WON'T) compile properly, because _____

_____.

b. (*4 pts.*) The Java `String` class has two different methods, both called `substring()`. Both are non-`void` methods that return _____.

The two versions of the method are differentiated by their input parameters. One version has _____ input parameters , and produces a substring that _____

_____.

The other version has _____ input parameters, and produces a substring that _____

_____.

3. **(10 pts.) CODE EVALUATION (I).**

For each of the following, give the value of the variable `x` after each line executes. If the line produces an error, then write `ERROR`. If the variable can have different values (as when using a random number generator), then indicate those values by writing, e.g., `1 <= x <= 5`.

a.
```
int x = 5 - 3 / 4 * 2;
```

b.
```
int x = (int)( 5 - 3 / 4.0 * 2 );
```

c.
```
int x = ( 5 - (int)( 3 / 4.0 ) ) * 2;
```

d.
```
double x = 5 - 3 / 4 * 2;
```

e.
```
String s = "Banana!";
char x = s.charAt( 3 );
```

f.
```
String s = "Banana!";
String x = s.substring( 1, 4 );
```

g.
```
String s = "Banana!";
String x = s + " has length " + s.length();
```

h.
```
int x = (int)( Math.random() * 10 + 10 );
```

i.
```
int x = (int) Math.random() * 10 + 10;
```

j.
```
boolean x = ( 8 / 2.0 == 4 );
```

4. **(4 pts.) CODE EVALUATION (II).**

Consider the following code:

```
String s1 = "This";
String s2 = "Is";
String s3 = "A";
String s4 = "Test";

s1 = s3;
s3 = s4;
s2 = s4;
s4 = s1;
```

---

a. The first four lines of code above create 4 distinct `String` objects in memory. When the code is complete, ***two*** of those objects have been orphaned. Write down the lines of code that causes this to happen, and write down the `String` literals that are orphaned by each.

b. Circle any method call below that would produce the result `true`, if executed ***after all*** of the code given in the question is complete. Remember that the `equals()` method returns `true` if and only if both the `String` that is calling the method and the input `String` are the same, character for character:

(i) `s1.equals( s3 )`

(ii) `s3.equals( s4 )`

(iii) `s2.equals( s4 )`

(iv) `s4.equals( s1 )`

5. **(10 pts.) CODE COMPLETION (I).**

On the next page, fill in the class given so that it contains a `main()` method that:

a. Uses `System.out` to ask the user for an integer value from `1` to `100` (inclusive), and reads it in from `System.in`, using a `Scanner`.

b. Generates its own random number from `1` to `100` (inclusive), and tells the user if their guess was correct, too high, or too low, while also showing the random number that was generated whenever they guess incorrectly. (See below for expected format.)

c. Tells the user that they have made an improper guess if they either (i) enter an integer value that is greater than `100` or less than `1`, or (ii) enter something that is not in proper integer format, like a floating-point number of text string.

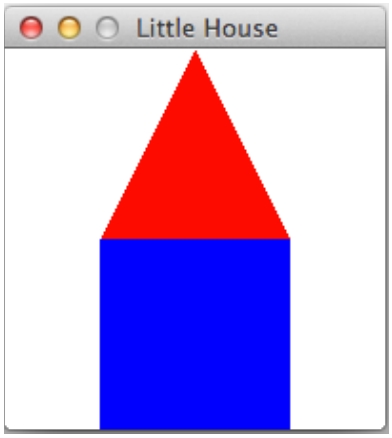
Thus, six different runs of the program—the first three with correct input, and the others with incorrect input—could be:

```
Please guess an integer between 1 and 100 (inclusive): 50
You guessed too high (correct answer: 14).

Please guess an integer between 1 and 100 (inclusive): 50
You guessed too low (correct answer: 78).

Please guess an integer between 1 and 100 (inclusive): 50
You guessed right!

Please guess an integer between 1 and 100 (inclusive): 210
That is an improper guess...

Please guess an integer between 1 and 100 (inclusive): 3.5
That is an improper guess...

Please guess an integer between 1 and 10 (inclusive): flapjacks
That is an improper guess...
```

```
// write the code for Question 5 here
import java.util.Scanner;

public class Q5
{









}
```

6. **(10 pts.) CODE COMPLETION (II).**

On the next page, complete the class given so that it creates something like the following image when run (although this is printed in black and white, here we are seeing a red triangle on top of a blue square).



a. Set the size of the window at random, so it is a square of side-length L, with $200 \leq L \leq 500$.

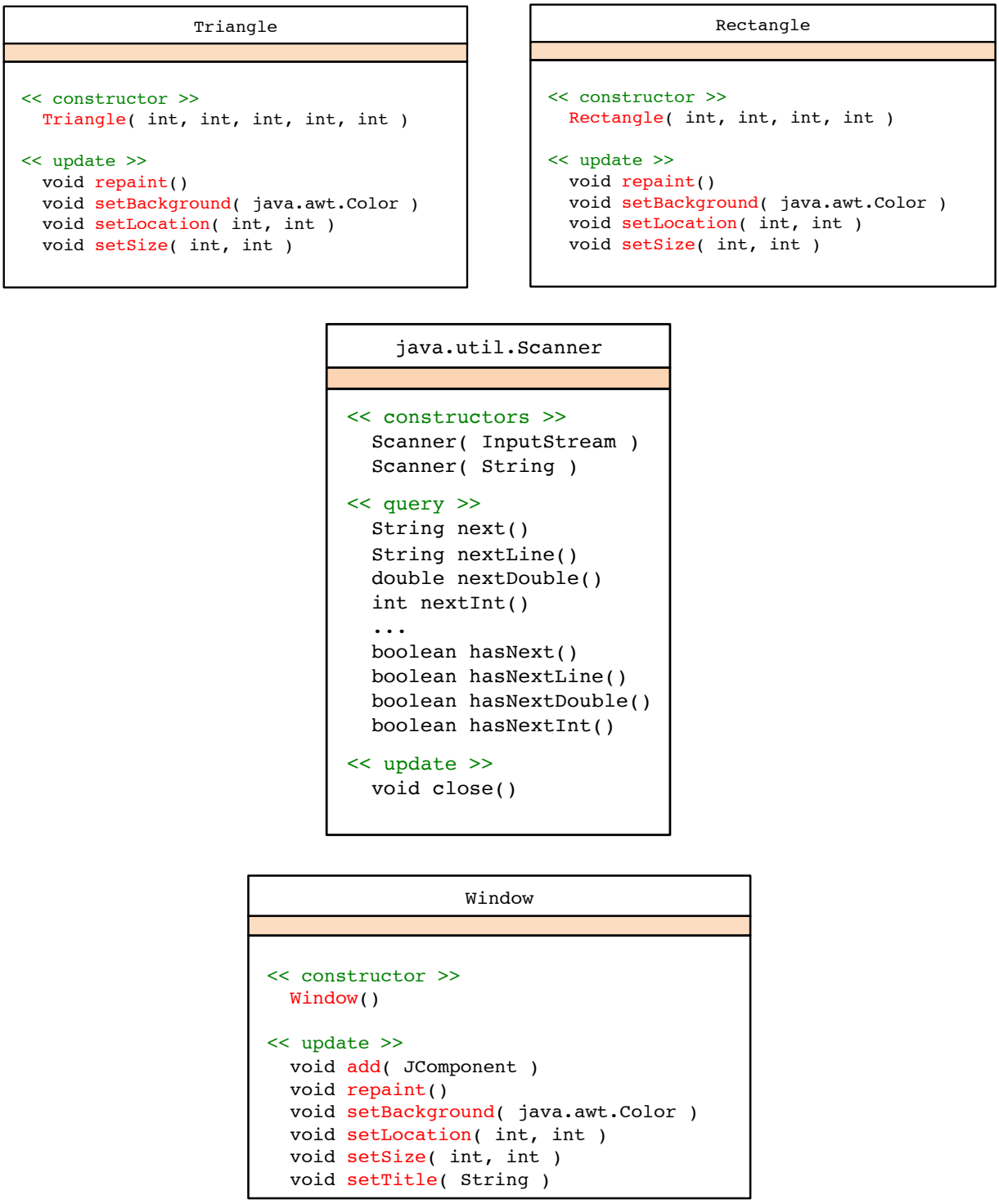
b. Create two objects, a `Rectangle` and a `Triangle`; each should have a side-length that is ***one half*** that of the size of the window; i.e., if the window itself is of size (300×300), then each of the two shapes should be of size (150×150). They should be arranged so that they are centered horizontally in the window; in addition, the square should be placed at the bottom of the window, with the triangle directly on top of the square.

c. Choose a random color combination for the two objects. Each should be either blue or red, and each of the four possible combinations (both blue, both red, or one of each color) should occur with equal probability on any given run of the program.

Note 1: Class diagrams for required graphical classes appear on the last page of the exam.

Note 2: The last input to the `Triangle()` constructor gives its orientation: if this number is `1`, then it will point upwards, and for any other value it will point downwards.

```
// Write the code for Question 6 here.
import java.awt.Color;

public class Q6
{
    public static void main( String[] args )
    {
        Window win = new Window();
        win.setTitle( "Little House" );
        win.setBackground( Color.white );









    }
}
```


Triangle
<< constructor >>
Triangle(int, int, int, int, int)
<< update >>
void repaint()
void setBackground(java.awt.Color)
void setLocation(int, int)
void setSize(int, int)
Rectangle
<< constructor >>
Rectangle(int, int, int, int)
<< update >>
void repaint()
void setBackground(java.awt.Color)
void setLocation(int, int)
void setSize(int, int)
java.util.Scanner
<< constructors >>
Scanner(InputStream)
Scanner(String)
<< query >>
String next()
String nextLine()
double nextDouble()
int nextInt()
...
boolean hasNext()
boolean hasNextLine()
boolean hasNextDouble()
boolean hasNextInt()
<< update >>
void close()
Window
<< constructor >>
Window()
<< update >>
void add(JComponent)
void repaint()
void setBackground(java.awt.Color)
void setLocation(int, int)
void setSize(int, int)
void setTitle(String)