

UNIVERSITY OF WISCONSIN-LA CROSSE
Department of Computer Science

CS 120
Midterm Exam 01

Software Design I

Fall 2016
11 October 2016

- 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 a **double** variable, then any numbers in an expression on the right-hand side will always be treated as of **double** type.

- 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. When doing arithmetic in Java, cast operations have higher precedence than any mathematical operations (multiplication, division, addition, subtraction and remainder).

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

```
int x = 1 / 2;
```

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

```
double x = 1 / 2;
```

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

```
String x = 1 / 2;
```

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

```
if( x < 0 )  
    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)( 3.0 / 2.0 );
```

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

_____.

An assignment like the following:

```
int x = (int) 3.0 / 2.0;
```

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

_____.

- b. (4 pts.) When creating an identifier for a variable, class, or method, there are certain rules

that we must follow. An identifier ***must*** begin with _____ ,

followed by _____ .

Other rules of style are not strictly enforced, but help make our code more readable. One

rule is that a class name ***should*** begin with _____ ,

while variables or method names ***should*** begin with _____ .

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 = 7 - 3 / 2 * 2;`

b. `int x = (int)(7 - 3 / 2.0 * 2);`

c. `int x = (7 - (int)(3 / 2.0)) * 2;`

d. `double y = 7 - 3 / 2 * 2;`

e. `String s = "Banana!";`
 `char x = s.charAt(s.length() - 1);`

f. `String s = "Banana!";`
 `String x = s.substring(2, 5);`

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

h. `int x = (int)(Math.random() * 100 + 1);`

i. `int x = (int) Math.random() * 100 + 1;`

j. `boolean x = (9 / 2 != 8 / 2);`

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

Consider the following code:

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

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

- a. The first four lines of code above create 4 distinct **String** objects in memory. When the code is complete, one of those objects have been orphaned. Write down the line of code that causes this to happen, and write down the **String** that is orphaned by it.
- 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:

(i) `s1.equals(s2)`

(ii) `s2.equals(s3)`

(iii) `s1.equals(s4)`

(iv) `s2.equals(s4)`

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, and reads it in from `System.in`, using a `Scanner`.
- b. If the user enters something other than an integer, the program should print out a message (as seen below) telling the user that the input is improper.
- c. Otherwise, the program should generate a random integer value from 1 to 5 (inclusive), and tell the user if their guess was correct, too high, or too low, while also showing the random number whenever they guess incorrectly. (See below for expected format.)

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

```
Please guess an integer: 3
You guessed too high (correct answer: 1).
```

```
Please guess an integer: 3
You guessed too low (correct answer: 4).
```

```
Please guess an integer: 3
You guessed right!
```

```
Please guess an integer: 3.5
That is an improper guess...
```

```
Please guess an integer: flapjacks
That is an improper guess...
```

Note: A partial class diagram for the `Scanner` class is found on the last page of this exam.

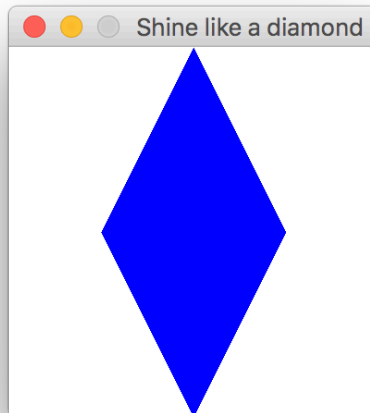
```
// write the code for Question 5 here
import java.util.Scanner;
```

```
public class Q5
{
```

```
}
```

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

On the next page, complete the given **Driver** class so that it can create something like the following image when run (although this is printed in black and white, here we are seeing a blue diamond, created out of two blue triangles).



- Create two **Triangle** objects; each should be of size (100×100) pixels. They should be arranged so that they are centered horizontally in the window, with one directly on top of the other, pointing up and down respectively.
- Choose a random color for the two objects. Each time the program runs the two shapes should both be either red, blue or green, and 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( "Shine like a diamond" );
        win.setBackground( Color.white );
        int winSize = 200;
        win.setSize( winSize, winSize );

    }
}
```

java.util.Scanner
<pre> << 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() </pre>

Window
<pre> << constructor >> Window() << update >> void add(JComponent) void repaint() void setBackground(java.awt.Color) void setLocation(int, int) void setSize(int, int) void setTitle(String) </pre>

Triangle
<pre> << constructor >> Triangle(int, int, int, int, int) << update >> void repaint() void setBackground(java.awt.Color) void setLocation(int, int) void setSize(int, int) </pre>