NAME: ________________________________

- 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.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>
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 you do not `import` a class for use in your code, then you cannot create or otherwise use objects of that class.

b. Within a single code block, no object can have two different identifiers.

c. Within a single code block, no two objects can be declared with the same identifier.

d. In a while loop with index variable `i`, the loop body must always include a line that increments (adds 1 to) the value of `i`.

e. Variables can be named anything you like, without exception.

f. The following line of code will not compile:

```
int x = (int) 20 * 3.3;
```

g. The following line of code will not compile:

```
int x = 20 * (int) 3.3;
```

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

```java
if( x < 0 )
    System.out.println( "One" );
else if ( x != 0 )
    System.out.println( "Two" );
```
2. (8 pts.) SHORT ANSWER.

a. (2 pts.) How many times will the following loop run, assuming it is in a correct program? (This is the same as the number of lines of output each produces.)

```java
int num = 0;
while ( num < 11 ) {
    System.out.println( num );
    num = num + 1;
}
```

Answer: ____________________________

b. (3 pts.) When we want to convert a more-precise primitive type to a less-precise type in Java, we must use ____________________________.

An example is ____________________________.

c. (3 pts.) Give three examples of errors in your code that will prevent the code from compiling.

1. ____________________________

2. ____________________________

3. ____________________________
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 = 3 / 2 * 4 + 6;

b. int x = (int)( 3 / 2.0 * 4 + 6 );

c. int x = 3 / (int)( 2.0 * 4 + 6 );

d. double x = 10 / 4 + 11;

e. double x = 10.0 / 4 + 11;

f. String x = "num = " + (3 + 6);

g. String x = "num = " + 3 + 6;

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

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

j. boolean x = ( 2 != 2.0 );
4. (4 pts.) CODE EVALUATION (II).

Consider the following code:

```java
Oval o1, o2, o3, o4;
o1 = new Oval( 50, 50, 100, 100 );
o2 = new Oval( 100, 100, 200, 200 );
o3 = null;
o4 = new Oval( 200, 200, 300, 300 );

o1 = o2;
o2 = o3;
o3 = o4;
o4 = o1;

o1.setBackground( Color.blue );
o2.setBackground( Color.red );
o3.setBackground( Color.green );
o4.setBackground( Color.magenta );
```

a. When this code is complete, two of the Oval variable identifiers refer to the same object in memory. What are those two identifiers?

b. When this code completes, one of the Oval objects will no longer have an identifier. Write down the code line that has this effect.
5. (10 pts.) CODE COMPLETION (I).

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

a. Asks the user for an integer value via `System.out`, and reads it in from `System.in`, using a `Scanner`.

b. Displays the **absolute value** of that input, so that if the user enters a negative number, it displays it in positive form. (See below for required format.)

c. Displays the **cube** of the value, so that if the user enters a number \( n \), it will display the value of \( n^3 \).

d. Treats the required input value as zero if it is in incorrect form.

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

```
Please enter an integer value: -5
Absolute value: 5
Cube: -125

Please enter an integer value: 5
Absolute value: 5
Cube: 125

Please enter an integer value: banana
Absolute value: 0
Cube: 0
```
// write the code for Question 5 here
import java.util.Scanner;

public class Q5
{
}
6. (10 pts.) CODE COMPLETION (II).

Complete the given class so that it can execute the following steps (continue on the next page if you need more space):

a. Create two different random integer values that are either 1 or 2.

b. If the first of the two numbers is less than the other, then a circle with diameter of 50 pixels is placed in the window, centered vertically and horizontally.

c. If the first of the two numbers is greater than the other, then a square with sides of 50 pixels is placed in the window, centered vertically and horizontally.

d. If the two numbers are the same, the background of the window is turned black.

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

```java
import java.awt.Color;

public class Q6
{
    public static void main( String[] args )
    {
        Window win = new Window();
        int winSize = 300;
        win.setSize( winSize, winSize );
    }
```
Oval

<< constructor >>
Oval( int, int, int, int )

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

<< query >>
java.awt.Color getBackground()

Rectangle

<< constructor >>
Rectangle( int, int, int, int )

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

Triangle

<< constructor >>
Triangle( int, int, int, int, int )

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

Window

<< constructor >>
Window()

<< update >>
void add( JComponent )  
void repaint()  
void setBackground( java.awt.Color )  
void setLocation( int, int )  
void setTitle( String )