- **CODE EVALUATION.**

  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., `x == 1 - 5`.

  i. `int x = (int) 10 / 2.5 * 2 + 5;`

  ii. `int x = (int)( 10 / 2.5 * 2 + 5 );`

  iii. `int x = 10 / (int) 2.5 * 2 + 5;`

  iv. `int x = 10 / (int)( 2.5 * 2 + 5 );`

  v. `double x = 12 / 5 + 11;`

  vi. `double x = 12.0 / 5 + 11;`

  vii. `int x = (int)( Math.random() * 10 ) - 10;`

  viii. `boolean x =  !( 3 < 6 < 5 );`

  ix. `boolean x = !( 3 < 6 && 6 < 5 );`

  x. `String x = 4 + " " + 5;`

  xi. `String x = "" + 4 + 5;`

- **CODE ERROR ANALYSIS**

  The following code contains **5 errors**. Circle each one. Below the code, describe what each error actually is; you can use the line-numbers in the code to identify each error if you like. Pay attention to the comments before the method: **it counts as an error if the code does not do what is described there.**

```
 1.   // Generates a random number from 1 to 10; then prints
 2.   // the number, followed by whether it is even or odd.
 4.   public static void main( String[] args )
 5.   {
 6.      int num = (int)( Math.random() * 10 );
 7.      if ( num % 2 == 1 );
 8.      {
 9.          System.out.println( "num is odd" );
10.      }
11.      else if ( num % 2 != 0 )
12.      {
13.          System.out.println( num + " is even" );
14.   }
```

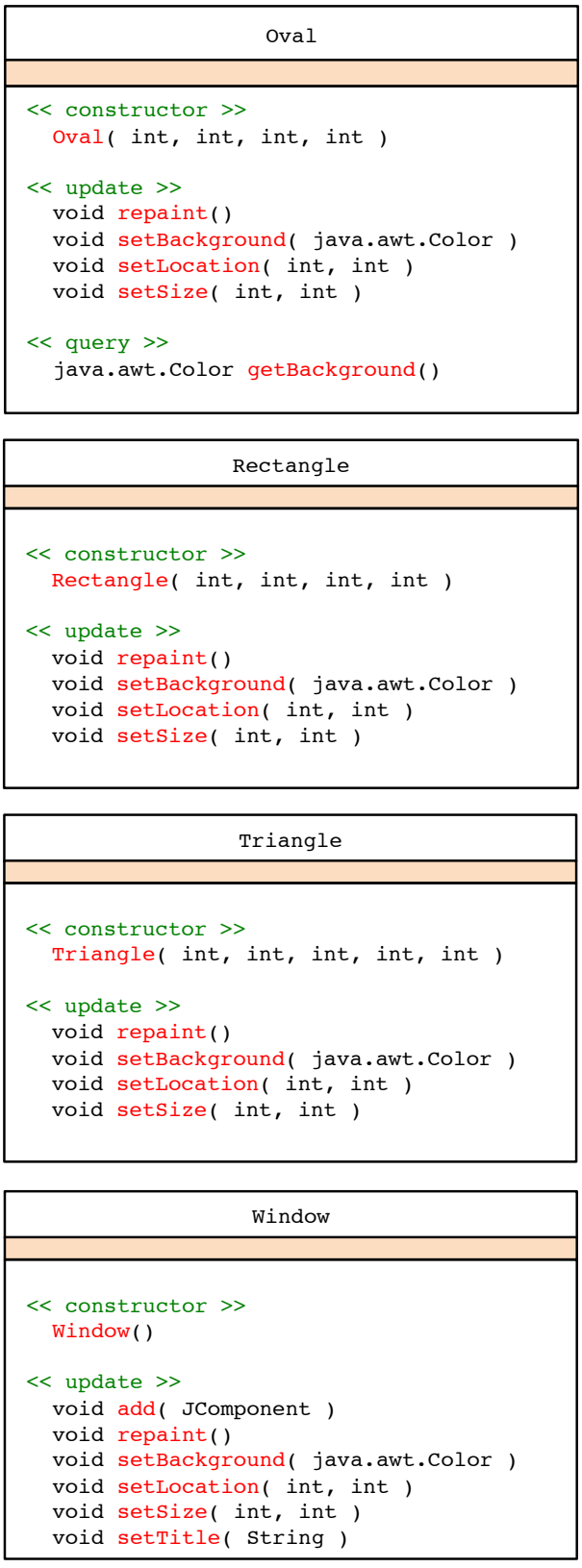
- **CODE COMPLETION.**

  On the next page, complete the given `Driver` class so that it can do the following.

  a. Create a `Window`: it should be at (x, y) location $(100, 100)$, with title "My Window" and background color **white**.
  b. Set the `Window` to have a **randomly selected size**, between `200` pixels minimum and `800` pixels maximum. It should be square.
  c. Create a shape object. This shape should be randomly chosen to either be an `Oval` or a `Rectangle`, with even probability, and it should be **blue**.
  d. Choose a **random size** for your shape object: it should be between `100` pixels minimum and the size of the containing `Window` at maximum. The size should be used for both width and height of the shape, so that it will either be a circle or a square.
  e. Place the shape in the `Window` so that it is centered (this will require calculations based on the size of both the shape and the `Window` itself).

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

```
// complete the following class to meet the requirements on the previous page


public class Driver
{









}
```


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 setSize(int, int)
void setTitle(String)