RIT | Golisano College of Computing and Information Sciences
School of Information

Name: ________________________________

ISTE-120
Lab 01: Using VSCODE

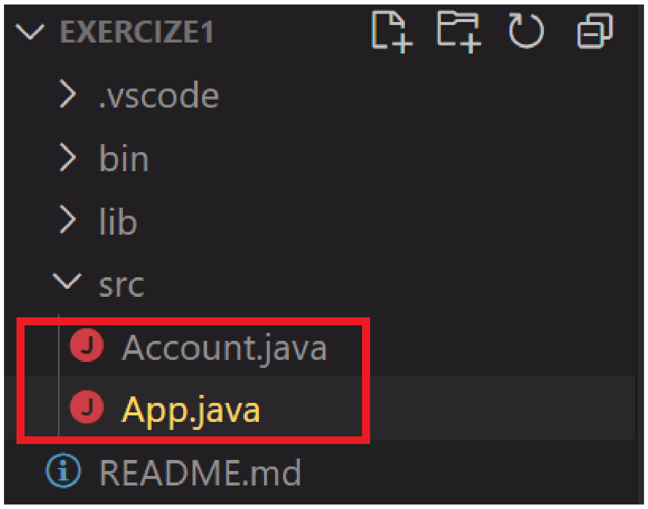
This assignment relates and contributes to the course learning outcomes: CLO1

Exercise 1 – Using an existing class (1 point)

This exercise must be completed during the lab period.

Create a new class

1. Download Lab01Downloads.zip from myCourses
2. Extract the Lab01Downloads.zip file to your computer
3. Make a working directory. For example, on a flash drive, you might have a directory for ISTE-120. Create a directory/folder under that and call it Lab01.
4. In your working directory, create a new VSCODE Project called Exercise1
5. Copy **Account.java** from the downloads to the Exercise1\src folder
6. Both files (Account.java and App.java) appear in the VSCODE



Signature: ______________________________ Date: _____________

Have the instructor or TA sign here when Exercise 1 compiles correctly.

Exercise 1 (Continued…) – Creating instances of a class (that is, objects) (3 points)

This exercise must be completed during the lab period.

A class is used as a template to create multiple **objects**. Each **object** is called an instance. An instance/object is created by running the class's constructor method via the new statement.

1. In VSCODE, open App.java file. Rename it to **Lab01.java**.

2. In the main program (public static void main…) , add code to create an instance of the Account class, called **account1**:
```
Account account1 = new Account();
```
3. In a similar way, create another instance of **Account** called **account2.**
4. "Call" the initialize "method" for the **account1** object.
```
account1.initialize("Jane");
```
   The item inside the parentheses is called a "parameter".
5. In a similar way, call the **deposit** method of **account1** and deposit `$100` into the account. The amount to deposit should be the parameter to **deposit**.
6. Now run the **withdraw** method of **account1** and withdraw `$40`. This time, the parameter should be 40.
7. Similar to the above sequence for **account1**, initialize the **account2** owner to `"Fred"` and deposit `$200` into **account2**. Do not withdraw anything at this time.

Printing out the object attributes:

The state of an object (instance) is determined by the values of its attributes. In the case of these accounts, the attributes are owner and balance.

1. After account1 has been initialized in the main, call the **print** method for it (there are no parameters to the **print** method). Precede the output of the print method with the label "Account1: " (note the space after :) as in:
```
// Note: using print, instead of println, cause the
// following two lines to be printed on one line
System.out.print("Account1: ");
account1.print();
```
   Insert the above statements after $100 is deposited in **account1**.

   Repeat this code again, after $40 is withdrawn.

   Run the program at this time to see how **account1** changes. Look for the output in the VSCODE Terminal window.

2. In the same way print the attributes of **account2**, with a label, after it is initialized.
3. **Print** the attributes of **account2** after the **deposit** and note the change in the balance. Methods, such as **deposit()** and **withdraw()**, give the object behavior and often change the state of the object by changing the values of its attributes. ALSO note that these methods must be called with the name of the object on the left side of a dot (.) and the name of the method on the right side. This is so Java knows which object is being accessed.
4. Run the **withdraw()** method of **account2** and withdraw `$125` as in

   ```
   account2.withdraw(125);
   ```

   Print the attributes of **account2** again to see the change.

Continue manipulating the objects and use the print their attributes to examine the state of the object after each action:

1. Create another account called **account3**.
2. Set the owner of **account3** to be `"George"`.
3. Deposit `$50` into **account3**.
5. Deposit `$60` into **account2.**
6. Withdraw `$20` from **account1.**
7. Complete the following table for the three Account objects:

Account Name	Owner	Balance
account1	Jane	
account2	Fred	
account3	George	

Signature: ______________________________ Date: _____________

Obtain instructor or TA signature.

Exercise 2 – Using classes and objects in VSCODE (3 points)

The exercise must be completed during the lab period.

A. Starting a new drawing project

1. Create a new VSCODE Project called Exercise2
2. Download the Drawing.jar library from myCourses (See Drawing.jar in Content → Drawing). Put it where you can find it easily (the VSCODE projects' ROOT directory)
3. Add it to the Exercize2 project (see VSCODE&JAVA_How_TO.pdf)
4. In VSCODE, open App.java file. Rename it to FirstDrawing.java

B. Creating Objects and Invoking Methods

1. Create a new Canvas window, change the NAME to your name. Width and Height are set to 500.

   Canvas canvas = new Canvas("FirstDrawing - NAME", 500, 500);

2. Create a Circle (called myCircle) with its upper left corner at (20, 60) and with a radius of 30. Print out its attributes with the statement:
   ```
   System.out.println("My Circle: " + myCircle);
   ```
   Printing a circle this way prints the string that is returned by calling the toString method. So the above statement is the same as:
   ```
   System.out.println("My Circle: " + myCircle.toString());
   ```
   Try this to check it out. Also, tell the canvas to draw the circle:
   ```
   canvas.draw(myCircle);
   ```
2. Create a new blue circle (myBlueCircle) at the same position as the myCircle(in the **new** statement add Color.BLUE, remember to import "java.awt.*;" as the first line of your program). Draw the blue circle, print the values and run the program again to see that it works.
   Be sure you understand the output of the print statement for myBlueCircle.
3. What is the blue circle's (x, y) position? (____, ____)
4. Now, before the code that <u>prints and draws</u> myBlueCircle, and after the **new** statement, insert the code:
   ```
   myBlueCircle.setXInt(myBlueCircle.getXInt() + 150);
   ```
   Note the X and the I in getXInt are capitalized. Now, rerun the program.

   What is the circle's new (x, y) position? (____, ____)

   getXInt returns the x position of the circle

   setXInt changes the x position of the circle

   Comment out the code that was added above:
   ```
   myBlueCircle.setXInt(myBlueCircle.getXInt() + 150);
   ```
   Add the code the following after the commented out code:

```
    myBlueCircle.setYInt(myBlueCircle.getYInt() + 200);
```

What should the (x, y) values be if this is run now? *** DO NOT RUN THE CODE *** INSTEAD PREDICT WHAT WILL HAPPEN.

(x=____, y=____)

NOW RUN THE CODE. What were the actual the (x, y) values when you ran your code with this change? (x=____, y=____). Was your prediction correct?

5. What code would you use to move the blue circle 7 pixels to the LEFT?

____

6. What code would you use to move the blue circle 19 pixels UP?

____

Signature: ____________________________ Date: ____________

Have the instructor or TA sign here when Exercise 3 works correctly.

©Peter Lutz, David Patric 2020

Exercise 3 – Drawing a Picture with Java (3 points)

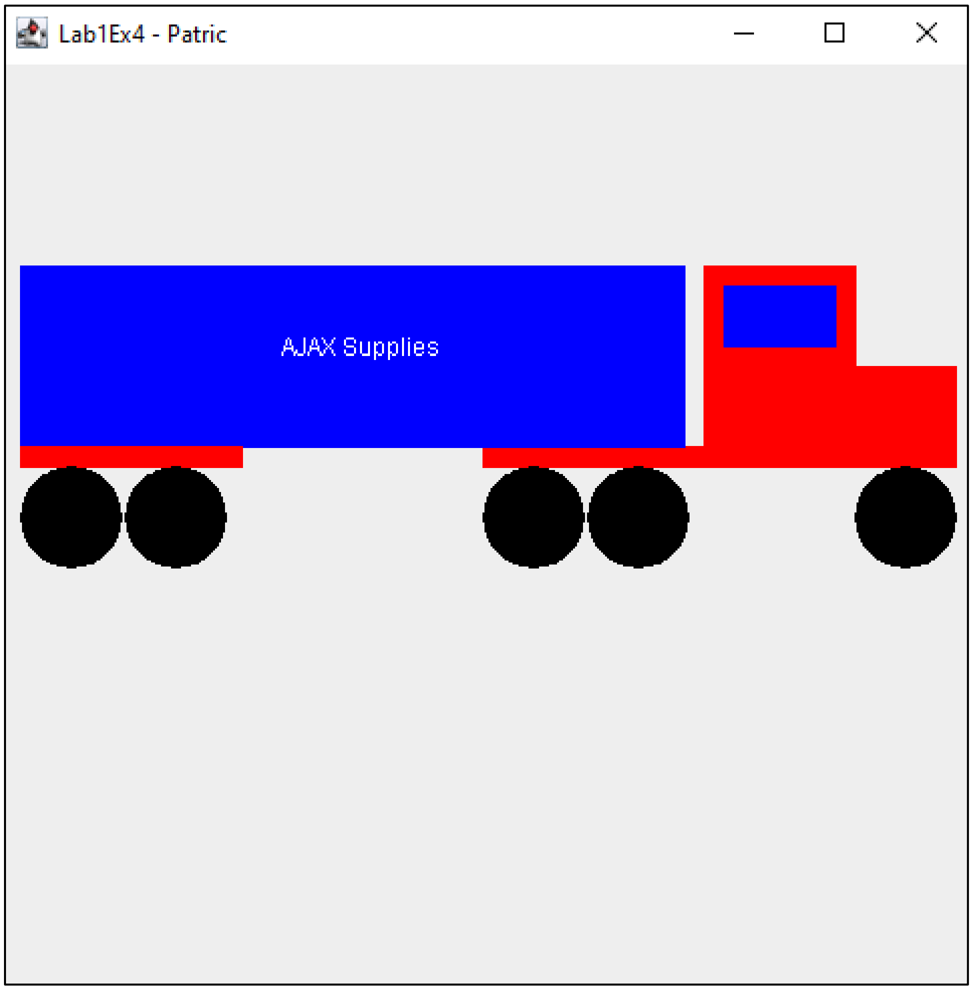
If this exercise is not completed during the lab period, complete the work outside of the lab period and bring the completed work to the lab next week.

- Create a new VSCODE Project called Exercise3
- Add Drawing.jar to the Exercize3 project (see VSCODE&JAVA_How_TO.pdf)
- In VSCODE, open App.java file. Rename it to TruckDrawing.java

- Create a new Canvas window, change the NAME to your actual name.

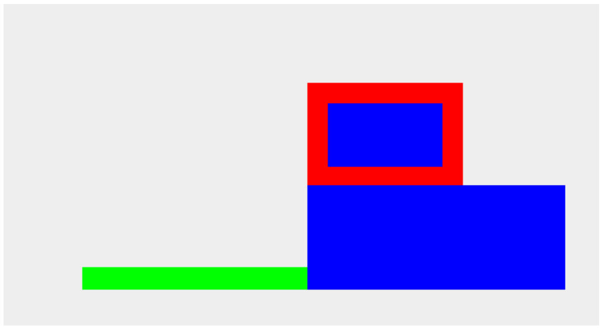
```
Canvas canvas = new Canvas("TruckDrawing - NAME", 500, 500);
```

Write a Java program to produce the following picture.



Hints:

This is drawn on a 500x500 canvas
The wheels all use the same Y value, width, and length … only the X value changes
The trailer (the blue part) is one big rectangle
The tractor (cab) is 4 rectangles, shown below in different colors (so you can see the 4 rectangles)



The carriage (for the rear-most wheels) is a shorter version of the green rectangle shown above, but with some modifications:

The width is a little less, but with the same y position and height, and moved to the left (same x position as the trailer).

Signature: _______________________________ Date: ______________

Have the instructor or TA sign here when Exercise 3 works correctly.

©Peter Lutz, David Patric 2020